EPSON
EXCEED YOUR VISION

CMOS 16-BIT SINGLE CHIP MICROCOMPUTER

S1C17701
TECHNICAL MANUAL

SEIKO EPSON CORPORATION

NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of Economy, Trade and Industry or other approval from another government agency.

This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

S1C17701 Technical Manual Revision History

Code No.	Page	Chapter/Section	Contents
411089901	1-1	1 Overview	Table 1.1 deleted. Description added. Main (OSC3) oscillator • Crystal/ceramic oscillator 8.2 MHz (max.) • CR oscillator 2.2 MHz (max.)
	1-2	1.1 Features	Part numbers deleted. Main (OSC3) oscillator (• S1C17701F00B100) (• S1C17701F00E100)
			Descriptions modified. Shipping form Part number for plastic package modified. Descriptions added. • VFBGA7H-161 package (7 mm × 7 mm × 1.0 mm, ball pitch: 0.5 mm) • VFBGA10H-144 package (10 mm × 10 mm × 1.0 mm, ball pitch: 0.8 mm)
	1-4	1.3.1 Pin Arrangement	Part number added. TQFP24-144-pin
			Figure 1.3.1.1 Part number modified. QFP24-144pin→TQFP24-144-pin
	1-5		Figure 1.3.1.2 added.
	1-6		Figure 1.3.1.3 added.
	1-7	1.3.2 Pin Description	Table 1.3.2.1 modified.
	1-8		Descriptions added. ∗1: SEG17 to SEG55(VFBGA7)...C6, D6, E6, A5, B5, C5, A4, D5, E5, C4, B4, A3, D4, B3, A2, C3, B2
	6-15	6.7 Details of Control Registers	Description deleted. (The interrupts can be used to clear standby mode even if the corresponding interrupt enable bit is set to disable interrupt.)
	10-11	10.8 Details of Control Registers	Description modified. The PxIN[7:0] bits correspond to the Px[7:0] ports respectively and the voltage level on the port pin is read out in the input mode...In the output mode, an indefinite value is read out.
	11-9	11.7 16-bit Timer Output Signal	Numerical value modified. Expression for I^{2}C
	22-3	22.3.2 Frame Signal	Description deleted. (see Table 22.3.1.)
	22-9	22.6.1 Turning Display On and Off	Table 22.6.1.1 modified.
			Description modified. DSPC[1:0] is reset to 0x0 (display off) after initial resetting...since switching to SLEEP mode with the LCD display left on will degrade the LCD.
	22-14	22.8 Details of Control Registers	Table modified. Description for 0x50a0: LCD Display Control Register (LCD_DCTL)
	22-15		Table 22.8.2 modified.
			Description modified. DSPC[1:0] is reset to 0x0 (display off) after initial resetting...since switching to SLEEP mode with the LCD display left on will degrade the LCD.
	26-5	26.5 Current	Table modified.
	26-7	26.6.3 External Clock Input AC Characteristics	Table modified.
	27-1	27 Package	Part number modified. QFP24-144pin package→TQFP24-144-pin package
	27-2		Figure added. VFBGA7H-161 Package
	27-3		Figure added. VFBGA10H-144 Package
	28-2	28.2 Pad Coordinates	Table modified. Coordinates modified.
	AP-18	Appendix A List of I/O Registers	Table for 0x50a0–0x50a6 modified.
	AP-36	Appendix D Precautions on Mounting	Description for Noise-Induced Erratic Operations modified.
411089902		Index	Index added.
	3-4	3.2.4 Access Control for the Flash Controller	Table 3.2.4.1 modified.

Configuration of product number

Devices

S1 C 17xxx F 00E1 00

Packing specifications

00 :	Besides tape & reel	
0A :	TCP BL	2 directions
0B :	Tape & reel	BACK
0C :	TCP BR	2 directions
0D :	TCP BT	2 directions
0E :	TCP BD	2 directions
0F :	Tape & reel	FRONT
0G :	TCP BT	4 directions
0H :	TCP BD	4 directions
0J :	TCP SL	2 directions
0K :	TCP SR	2 directions
0L :	Tape & reel	LEFT
0M :	TCP ST	2 directions
0N :	TCP SD	2 directions
0P :	TCP ST	4 directions
0Q :	TCP SD	4 directions
0R :	Tape & reel	RIGHT
99 :	Specs not fixed	

Specification

Package
[D: die form; F: QFP, B: BGA]

Model number

Model name
[C: microcomputer, digital products]

Product classification
[S1: semiconductor]

Development tools

S5U1 C 17000 H2 1 00

Packing specifications
[00: standard packing]

Version
[1: Version 1]

Tool type

Hx :	ICE
Dx :	Evaluation board
Ex :	ROM emulation board
Mx :	Emulation memory for external ROM
Tx :	A socket for mounting
Cx :	Compiler package
Sx :	Middleware package

Corresponding model number
[17xxx: for S1C17xxx]

Tool classification
[C: microcomputer use]

Product classification
[S5U1: development tool for semiconductor products]

S1C17701 Technical Manual

Appendix **AP**

– Contents –

1 Overview

The S1C17701 is a 16-bit MCU that features high-speed operation, low power consumption, small size, large address space, and on-chip ICE. The S1C17701 consists of an S1C17 CPU Core, a 64K-byte Flash memory, a 4K-byte RAM, serial interface modules (UART that supports high bit rate and IrDA 1.0, SPI and I^2C) for connecting various sensor modules, 8-bit timers, 16-bit timers, a PWM & capture timer, a clock timer, a stopwatch timer, a watchdog timer, 28 GPIO ports, an LCD driver with 56-segment × 32-common outputs and a voltage booster, a supply voltage detector, 32 kHz (typ.) and 8.2 MHz (max.) oscillators, and a voltage regulator for generating the 1.8 V internal voltage. The S1C17701 is capable of high-speed operation (8.2 MHz) with low operating voltage (1.8 V). Its 16-bit RISC processor executes one instruction in 1.5 clock cycles.

The S1C17701 also provides an on-chip ICE function that allows on-board erasing/programming of the embedded Flash memory, on-board debugging and evaluating the program by connecting the S1C17701 to the ICD Mini (S5U1C17001H) with only three wires. The S1C17701 is suitable for battery driven applications with sensor interfaces and up to 56 × 32-dot LCD display, such as remote controllers and sports watches.

The product lineup offers two S1C17701 models with a different main oscillator.

Main (OSC3) oscillator	• Crystal/ceramic oscillator 8.2 MHz (max.)
	• CR oscillator 2.2 MHz (max.)

∗ This product uses SuperFlash® Technology licensed from Silicon Storage Technology, Inc.

1.1 Features

The main functions and features of the S1C17701 are outlined below.

CPU	• Seiko Epson original 16-bit RISC CPU core S1C17
Main (OSC3) oscillator	• Crystal/ceramic oscillator 8.2 MHz (max.) • CR oscillator 2.2 MHz (max.)
Sub (OSC1) oscillator	• Crystal oscillator 32.786 kHz (typ.)
On-chip Flash memory	• 64K bytes (for instructions and data) • 1,000 erase/program cycles • Read/program protection • On-board programming by a debugging tool such as ICD Mini (S5U1C17701H) and self-programming by software control
On-chip RAM	• 4K bytes
On-chip display RAM	• 576 bytes
I/O ports	• Max. 28 general-purpose I/O ports (Pins are shared with the peripheral I/O.)
Serial interfaces	• SPI (master/slave) 1 ch. • I²C (master) 1 ch. • UART (115200 bps, IrDA 1.0) 1 ch. • Remote controller (REMC) 1 ch.
Timers	• 8-bit timer (T8F) 1 ch. • 16-bit timer (T16) 3 ch. • PWM & capture timer (T16E) 1 ch. • Clock timer (CT) 1 ch. • Stopwatch timer (SWT) 1 ch. • Watchdog timer (WDT) 1 ch. • 8-bit OSC1 timer (T8OSC1) 1 ch.
LCD driver	• 56 SEG × 32 COM or 72 SEG × 16 COM (1/5 bias) • Built-in voltage booster
Supply voltage detector (SVD)	• 13 programmable detection levels (1.8 V to 2.7 V)
Interrupts	• Reset • NMI • 16 programmable interrupts (8 levels)
Power supply voltage	• 1.8 V to 3.6 V (for normal (low-power) operation with the 1.8 V internal voltage) • 2.7 V to 3.6 V (for Flash erasing/programming with the 2.5 V internal voltage)
Operating temperature	• -20°C to 70°C
Current consumption	• SLEEP state: 1 µA typ. • HALT state: 2.6 µA typ. (32 kHz OSC1 crystal oscillator, LCD off) • Run state: 14 µA typ. (32 kHz OSC1 crystal oscillator, LCD off) 1800 µA typ. (8 MHz OSC3 ceramic oscillator, LCD off)
Shipping form	• TQFP24-144pin plastic package (16 mm × 16 mm × 1.0 mm, lead pitch: 0.4 mm) • VFBGA7H-161 package (7 mm × 7 mm × 1.0 mm, ball pitch: 0.5 mm) • VFBGA10H-144 package (10 mm × 10 mm × 1.0 mm, ball pitch: 0.8 mm) • Chip

1.2 Block Diagram

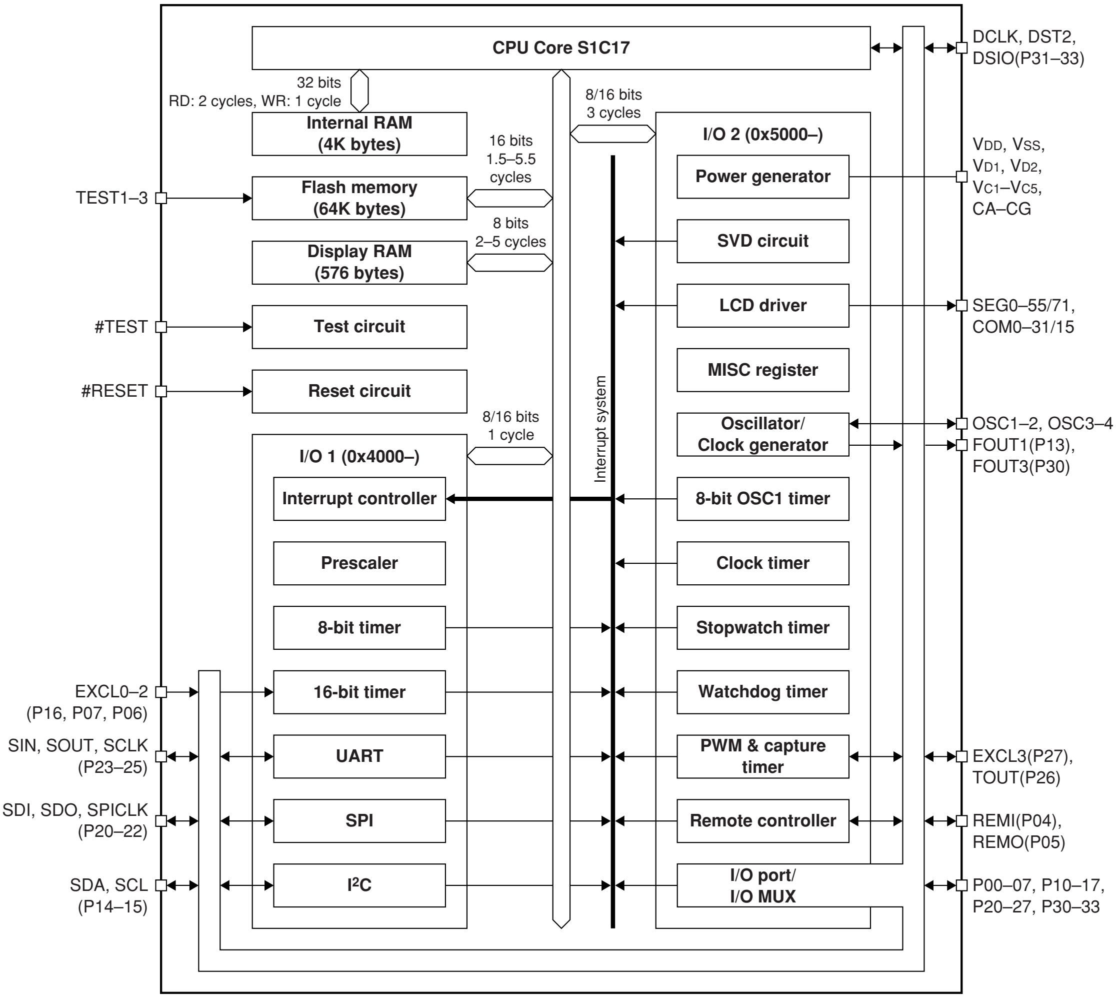


Figure 1.2.1 Block Diagram

1.3 Pins

1.3.1 Pin Arrangement

TQFP24-144-pin

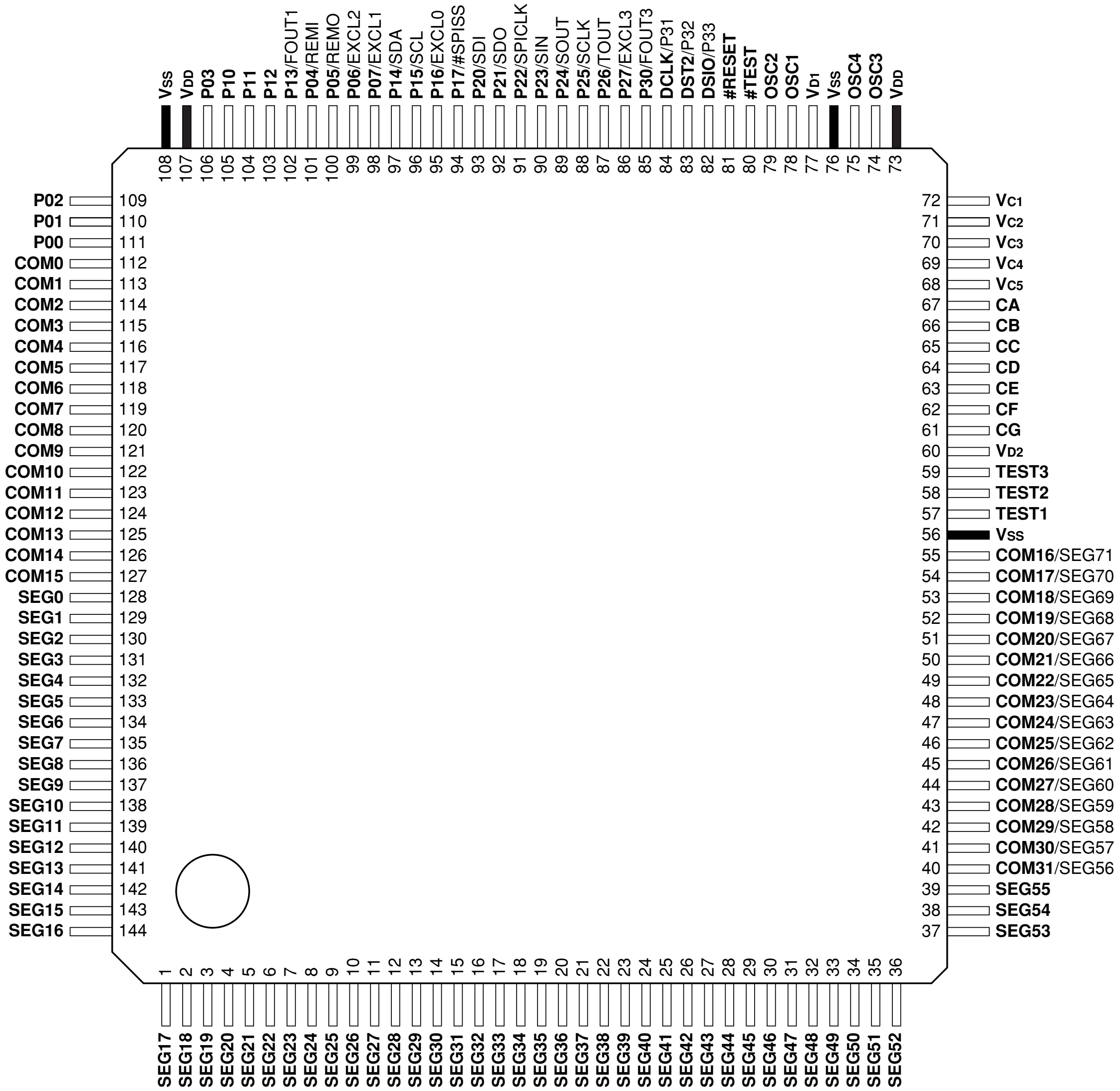
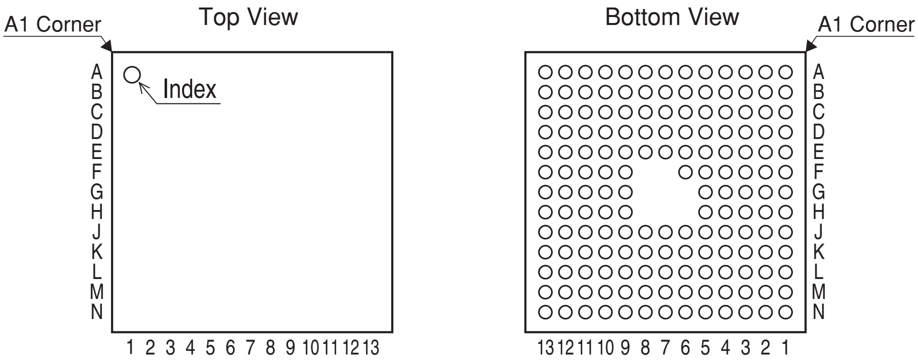


Figure 1.3.1.1 Pin Arrangement (TQFP24-144-pin)

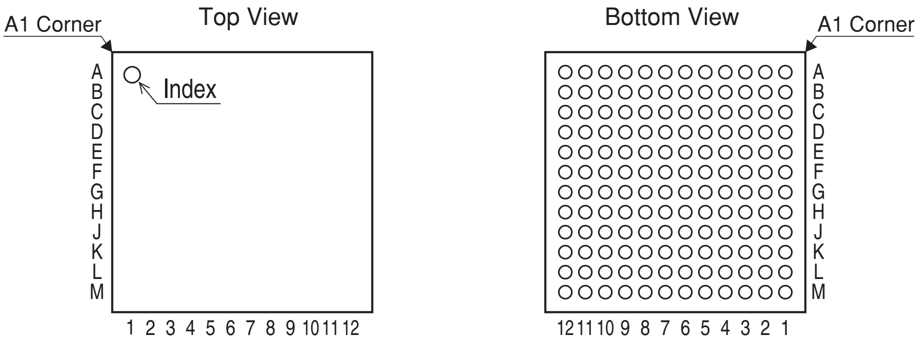
VFBGA7H-161



	1	2	3	4	5	6	7	8	9	10	11	12	13	
A	N.C.	SEG16	SEG12	SEG10	SEG7	SEG3	COM15	COM12	COM6	COM4	COM0	P01	N.C.	A
B	SEG17	SEG15	SEG13	SEG8	SEG6	SEG2	COM14	COM9	COM7	COM3	P00	P02	P03	B
C	SEG20	SEG18	SEG14	SEG11	SEG9	SEG4	COM13	COM11	COM8	COM2	P12	P11	P10	C
D	SEG23	SEG22	SEG21	SEG19	SEG5	SEG1	SEG0	COM10	COM5	COM1	P13 FOUT1	P04 REMI	P05 REMO	D
E	SEG27	SEG26	SEG25	SEG24	VSS	VSS	VSS	VSS	VDD	P06 EXCL2	P07 EXCL1	P14 SDA	P15 SCL	E
F	SEG30	SEG29	SEG28	SEG31	VSS	VSS			VDD	P16 EXCL0	P20 SDI	P21 SDO	P22 SPICLK	F
G	SEG33	SEG32	SEG34	SEG35	VSS		Top View		VSS	P17 #SPISS	P23 SIN	P24 SOUT	P25 SCLK	G
H	SEG37	SEG36	SEG38	SEG39	VSS				VSS	P27 EXCL3	P26 TOUT	P30 FOUT3	DCLK P31	H
J	SEG40	SEG41	SEG42	SEG43	VSS	VSS	VSS	TEST3	VDD	VC4	DST2 P32	#RESET	DSIO P33	J
K	SEG44	SEG45	SEG46	SEG48	COM28 SEG59	COM24 SEG63	TEST2	TEST1	CD	VC5	#TEST	OSC2	OSC1	K
L	SEG47	SEG49	SEG50	COM31 SEG56	COM27 SEG60	COM23 SEG64	COM20 SEG67	COM17 SEG70	CG	CA	VC1	VD1	VSS	L
M	SEG52	SEG51	SEG54	COM30 SEG57	COM26 SEG61	COM22 SEG65	COM19 SEG68	COM16 SEG71	CF	CC	VC2	OSC4	VDD	M
N	N.C.	SEG53	SEG55	COM29 SEG58	COM25 SEG62	COM21 SEG66	COM18 SEG69	VD2	CE	CB	VC3	OSC3	N.C.	N
	1	2	3	4	5	6	7	8	9	10	11	12	13	

Figure 1.3.1.2 Pin Arrangement (VFBGA7H-161)

VFBGA10H-144



Top View

	1	2	3	4	5	6	7	8	9	10	11	12	
A	SEG17	SEG14	SEG11	SEG06	SEG03	COM14	COM10	COM06	COM02	P00	P02	V_{SS}	A
B	SEG18	SEG16	SEG13	SEG10	SEG04	COM15	COM09	COM07	COM01	P01	P03	P10	B
C	SEG20	SEG19	SEG15	SEG09	SEG05	SEG00	COM08	COM05	COM00	P12	P11	P04 REMI	C
D	SEG23	SEG22	SEG21	SEG12	SEG07	SEG01	COM11	COM04	P06 EXCL2	P13 FOUT1	P05 REMO	P14 SDA	D
E	SEG27	SEG26	SEG25	SEG24	SEG08	SEG02	COM12	COM03	P07 EXCL1	P17 SPISS	P16 EXCL0	P15 SCL	E
F	SEG30	SEG29	SEG28	SEG31	SEG32	TEST2	COM13	V_{DD}	P23 SIN	P20 SDI	P21 SDO	P22 SPICLK	F
G	SEG34	SEG33	SEG35	SEG36	SEG37	TEST1	TEST3	P24 SOUT	P26 TOUT	P27 EXCL3	P25 SCLK	DCLK P31	G
H	SEG38	SEG42	SEG41	SEG40	SEG39	COM24 SEG63	V_{SS}	VDD TEST	P30 FOUT3	DST2 P32	RESET	DSIO P33	H
J	SEG43	SEG45	SEG44	SEG46	COM28 SEG59	COM23 SEG64	COM19 SEG68	CE	CD	V_{SS}	OSC2	OSC1	J
K	SEG47	SEG49	SEG48	COM31 SEG56	COM27 SEG60	COM22 SEG65	COM18 SEG69	CF	CC	V_{C5}	OSC4	V_{D1}	K
L	SEG50	SEG51	SEG54	COM30 SEG57	COM26 SEG61	COM21 SEG66	COM16 SEG71	CG	CB	V_{C4}	V_{C1}	OSC3	L
M	SEG52	SEG53	SEG55	COM29 SEG58	COM25 SEG62	COM20 SEG67	COM17 SEG70	V_{D2}	CA	V_{C3}	V_{C2}	V_{DD}	M
	1	2	3	4	5	6	7	8	9	10	11	12	

Figure 1.3.1.3 Pin Arrangement（VFBGA10H-144）

1.3.2 Pin Description

Table 1.3.2.1 Pin Descriptions

Pin No.			Name	I/O	Default status	Function
QFP	VFBGA7	VFBGA10				
1 to 39	*1	*2	**SEG17 to 55**	O	O(L)	LCD segment output pin
40 to 55	*3	*4	**COM31 to 16**/ SEG56 to 71	O	O(L)	LCD common output pin*/LCD segment output pin
56	*5	*6	**VSS**	–	–	Power supply pin (GND)
57	K8	G6	**TEST1**	–	–	Flash test pin (open in normal operation)
58	K7	F6	**TEST2**	–	–	Flash test pin (fixed at High in normal operation)
59	J8	G7	**TEST3**	–	–	Flash test pin (open in normal operation)
60	N8	M8	**VD2**	–	–	LCD circuit power supply booster output pin
61	L9	L8	**CG**	–	–	Power supply booster capacitor connector pin
62	M9	K8	**CF**	–	–	Power supply booster capacitor connector pin
63	N9	J8	**CE**	–	–	LCD booster capacitor connector pin
64	K9	J9	**CD**	–	–	LCD booster capacitor connector pin
65	M10	K9	**CC**	–	–	LCD booster capacitor connector pin
66	N10	L9	**CB**	–	–	LCD booster capacitor connector pin
67	L10	M9	**CA**	–	–	LCD booster capacitor connector pin
68	K10	K10	**VC5**	–	–	LCD circuit drive voltage output pin
69	J10	L10	**VC4**	–	–	LCD circuit drive voltage output pin
70	N11	M10	**VC3**	–	–	LCD circuit drive voltage output pin
71	M11	M11	**VC2**	–	–	LCD circuit drive voltage output pin
72	L11	L11	**VC1**	–	–	LCD circuit drive voltage output pin
73	*7	*8	**VDD**	–	–	Power supply pin (+)
74	N12	L12	**OSC3**	I	I	OSC3 oscillator input pin
75	M12	K11	**OSC4**	O	O	OSC3 oscillator output pin
76	*5	*6	**VSS**	–	–	Power supply pin (GND)
77	L12	K12	**VD1**	–	–	Internal logic and oscillator circuit constant-voltage circuit output pin
78	K13	J12	**OSC1**	I	I	OSC1 oscillator input pin
79	K12	J11	**OSC2**	O	O	OSC1 oscillator output pin
80	K11	H8	**#TEST** (VFBGA7) **VDD/#TEST** (VFBGA10)	I	I(Pull-UP)	Test pin (fixed at High in normal operation)
81	J12	H11	**#RESET**	I	I(Pull-UP)	Initial set input pin
82	J13	H12	**DSIO**/P33	I/O	I(Pull-UP)	On-chip debugger data input/output pin*/ input/output port pin
83	J11	H10	**DST2**/P32	I/O	O(L)	On-chip debugger status output pin*/ input/output port pin
84	H13	G12	**DCLK**/P31	I/O	O(H)	On-chip debugger clock output pin*/ input/output port pin
85	H12	H9	**P30**/FOUT3	I/O	I(Pull-UP)	Input/output port pin*/ OSC3 dividing clock output pin
86	H10	G10	**P27**/EXCL3	I/O	I(Pull-UP)	Input/output port pin*/T16E external clock input pin
87	H11	G9	**P26**/TOUT	I/O	I(Pull-UP)	Input/output port pin*/T16E PWM signal output pin
88	G13	G11	**P25**/SCLK	I/O	I(Pull-UP)	Input/output port pin*/UART clock input pin
89	G12	G8	**P24**/SOUT	I/O	I(Pull-UP)	Input/output port pin*/UART data output pin
90	G11	F9	**P23**/SIN	I/O	I(Pull-UP)	Input/output port pin*/UART data input pin
91	F13	F12	**P22**/SPICLK	I/O	I(Pull-UP)	Input/output port pin*/SPI clock input/output pin
92	F12	F11	**P21**/SDO	I/O	I(Pull-UP)	Input/output port pin*/SPI data output pin
93	F11	F10	**P20**/SDI	I/O	I(Pull-UP)	Input/output port pin*/SPI data input pin
94	G10	E10	**P17**/#SPISS	I/O	I(Pull-UP)	Input/output port pin (with interrupt)*/SPI slave select input pin
95	F10	E11	**P16**/EXCL0	I/O	I(Pull-UP)	Input/output port pin (with interrupt)*/T16 Ch.0 external clock input pin
96	E13	E12	**P15**/SCL	I/O	I(Pull-UP)	Input/output port pin (with interrupt)*/I^{2}C clock output pin
97	E12	D12	**P14**/SDA	I/O	I(Pull-UP)	Input/output port pin (with interrupt)*/I^{2}C data input/output pin
98	E11	E9	**P07**/EXCL1	I/O	I(Pull-UP)	Input/output port pin (with interrupt)*/T16 Ch.1 external clock input pin

Pin No.			Name	I/O	Default status	Function
QFP	VFBGA7	VFBGA10				
99	E10	D9	**P06**/EXCL2	I/O	I(Pull-UP)	Input/output port pin (with interrupt)*/T16 Ch.2 external clock input pin
100	D13	D11	**P05**/REMO	I/O	I(Pull-UP)	Input/output port pin (with interrupt)*/Remote output pin
101	D12	C12	**P04**/REMI	I/O	I(Pull-UP)	Input/output port pin (with interrupt)*/Remote input pin
102	D11	D10	**P13**/FOUT1	I/O	I(Pull-UP)	Input/output port pin (with interrupt)*/OSC1 clock output pin
103	C11	C10	**P12**	I/O	I(Pull-UP)	Input/output port pin (with interrupt)
104	C12	C11	**P11**	I/O	I(Pull-UP)	Input/output port pin (with interrupt)
105	C13	B12	**P10**	I/O	I(Pull-UP)	Input/output port pin (with interrupt)
106	B13	B11	**P03**	I/O	I(Pull-UP)	Input/output port pin (with interrupt)
107	∗7	∗8	**V_{DD}**	–	–	Power supply (+)
108	∗5	∗6	**V_{SS}**	–	–	Power supply (GND)
109	B12	A11	**P02**	I/O	I(Pull-UP)	Input/output port pin (with interrupt)
110	A12	B10	**P01**	I/O	I(Pull-UP)	Input/output port pin (with interrupt)
111	B11	A10	**P00**	I/O	I(Pull-UP)	Input/output port pin (with interrupt)
112 to 127	∗9	∗10	**COM0 to 15**	O	O(L)	LCD common output pin
128 to 144	∗11	∗12	**SEG0 to 16**	O	O(L)	LCD segment output pin

∗1: SEG17 to SEG55(VFBGA7)
B1, C2, D4, C1, D3, D2, D1, E4, E3, E2, E1, F3, F2, F1, F4, G2, G1, G3, G4, H2, H1, H3, H4, J1, J2, J3, J4, K1, K2, K3, L1, K4, L2, L3, M2, M1, N2, M3, N3

∗2: SEG17 to SEG55(VFBGA10)
A1, B1, C2, C1, D3, D2, D1, E4, E3, E2, E1, F3, F2, F1, F4, F5, G2, G1, G3, G4, G5, H1, H5, H4, H3, H2, J1, J3, J2, J4, K1, K3, K2, L1, L2, M1, M2, L3, M3

∗3: COM31 to 16/SEG56 to 71(VFBGA7)
L4, M4, N4, K5, L5, M5, N5, K6, L6, M6, N6, L7, M7, N7, L8, M8

∗4: COM31 to 16/SEG56 to 71(VFBGA10)
K4, L4, M4, J5, K5, L5, M5, H6, J6, K6, L6, M6, J7, K7, M7, L7

∗5: V_{SS}(VFBGA7)
E5, E6, E7, E8, F5, F6, G5, G9, H5, H9, J5, J6, J7, L13

∗6: V_{SS}(VFBGA10)
A12, H7, J10

∗7: V_{DD}(VFBGA7)
E9, F9, J9, M13

∗8: V_{DD}(VFBGA10)
F8, M12

∗9: COM0 to 15(VFBGA7)
A11, D10, C10, B10, A10, D9, A9, B9, C9, B8, D8, C8, A8, C7, B7, A7

∗10: COM0 to 15(VFBGA10)
C9, B9, A9, E8, D8, C8, A8, B8, C7, B7, A7, D7, E7, F7, A6, B6

∗11: SEG0 to 16(VFBGA7)
D7, D6, B6, A6, C6, D5, B5, A5, B4, C5, A4, C4, A3, B3, C3, B2, A2

∗12: SEG0 to 16(VFBGA10)
C6, D6, E6, A5, B5, C5, A4, D5, E5, C4, B4, A3, D4, B3, A2, C3, B2

Note: Bold text (for pins) and an asterisk (for functions) indicate default settings.

2 CPU

The S1C17701 contains the S1C17 Core as its core processor.
The S1C17 Core is a Seiko Epson original 16-bit RISC-type processor.
It features low power consumption, high-speed operation, large address space, main instructions executable in one clock cycle, and a small sized design. The S1C17 Core is suitable for embedded applications such as controllers and sequencers for which an eight-bit CPU is commonly used.
For details of the S1C17 Core, refer to the "S1C17 Family S1C17 Core Manual."

2.1 Features of the S1C17 Core

Processor type

- Seiko Epson original 16-bit RISC processor
- 0.35–0.15 µm low power CMOS process technology

Instruction set

- Code length: 16-bit fixed length
- Number of instructions: 111 basic instructions (184 including variations)
- Execution cycle: Main instructions executed in one cycles
- Extended immediate instructions: Immediate extended up to 24 bits
- Compact and fast instruction set optimized for development in C language

Register set

- Eight 24-bit general-purpose registers
- Two 24-bit special registers
- One 8-bit special register

Memory space and bus

- Up to 16M bytes of memory space (24-bit address)
- Harvard architecture using separated instruction bus (16 bits) and data bus (32 bits)

Interrupts

- Reset, NMI, and 32 external interrupts supported
- Address misaligned interrupt
- Debug interrupt
- Direct branching from vector table to interrupt handler routine
- Programmable software interrupts with a vector number specified (all vector numbers specifiable)

Power saving

- HALT (`halt` instruction)
- SLEEP (`slp` instruction)

2.2 CPU Registers

The S1C17 Core contains eight general-purpose registers and three special registers.

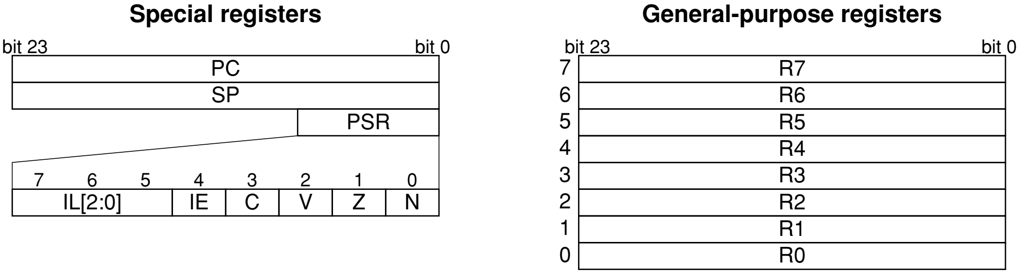


Figure 2.2.1 Registers

2.3 Instruction Set

The S1C17 Core instruction codes are all fixed to 16 bits in length which, combined with pipelined processing, allows most important instructions to be executed in one cycle. For details, refer to the "S1C17 Family S1C17 Core Manual."

Table 2.3.1 List of S1C17 Core Instructions

Classification	Mnemonic		Function
Data transfer	ld.b	%rd,%rs	General-purpose register (byte) → general-purpose register (sign-extended)
		%rd,[%rb]	Memory (byte) → general-purpose register (sign-extended) Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		%rd,[%rb]+	
		%rd,[%rb]-	
		%rd,-[%rb]	
		%rd,[%sp+imm7]	Stack (byte) → general-purpose register (sign-extended)
		%rd,[imm7]	Memory (byte) → general-purpose register (sign-extended)
		[%rb],%rs	General-purpose register (byte) → memory Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		[%rb]+,%rs	
		[%rb]-,%rs	
		-[%rb],%rs	
		[%sp+imm7],%rs	General-purpose register (byte) → stack
		[imm7],%rs	General-purpose register (byte) → memory
	ld.ub	%rd,%rs	General-purpose register (byte) → general-purpose register (zero-extended)
		%rd,[%rb]	Memory (byte) → general-purpose register (zero-extended) Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		%rd,[%rb]+	
		%rd,[%rb]-	
		%rd,-[%rb]	
		%rd,[%sp+imm7]	Stack (byte) → general-purpose register (zero-extended)
		%rd,[imm7]	Memory (byte) → general-purpose register (zero-extended)
	ld	%rd,%rs	General-purpose register (16 bits) → general-purpose register
		%rd,sign7	Immediate → general-purpose register (sign-extended)
		%rd,[%rb]	Memory (16 bits) → general-purpose register Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		%rd,[%rb]+	
		%rd,[%rb]-	
		%rd,-[%rb]	
		%rd,[%sp+imm7]	Stack (16 bits) → general-purpose register
		%rd,[imm7]	Memory (16 bits) → general-purpose register
		[%rb],%rs	General-purpose register (16 bits) → memory Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		[%rb]+,%rs	
		[%rb]-,%rs	
		-[%rb],%rs	
		[%sp+imm7],%rs	General-purpose register (16 bits) → stack
		[imm7],%rs	General-purpose register (16 bits) → memory
	ld.a	%rd,%rs	General-purpose register (24 bits) → general-purpose register
		%rd,imm7	Immediate → general-purpose register (zero-extended)
		%rd,[%rb]	Memory (32 bits) → general-purpose register (∗1) Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		%rd,[%rb]+	
		%rd,[%rb]-	
		%rd,-[%rb]	
		%rd,[%sp+imm7]	Stack (32 bits) → general-purpose register (∗1)
		%rd,[imm7]	Memory (32 bits) → general-purpose register (∗1)
		[%rb],%rs	General-purpose register (32 bits, zero-extended) → memory (∗1) Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		[%rb]+,%rs	
		[%rb]-,%rs	
		-[%rb],%rs	
		[%sp+imm7],%rs	General-purpose register (32 bits, zero-extended) → stack (∗1)
		[imm7],%rs	General-purpose register (32 bits, zero-extended) → memory (∗1)
		%rd,%sp	SP → general-purpose register
		%rd,%pc	PC → general-purpose register
		%rd,[%sp]	Stack (32 bits) → general-purpose register (∗1) Stack pointer post-increment, post-decrement, and pre-decrement functions can be used.
		%rd,[%sp]+	
		%rd,[%sp]-	
		%rd,-[%sp]	

Classification	Mnemonic		Function
Data transfer	ld.a	[%sp],%rs	General-purpose register (32 bits, zero-extended) → stack (*1) Stack pointer post-increment, post-decrement, and pre-decrement functions can be used.
		[%sp]+,%rs	
		[%sp]-,%rs	
		-[%sp],%rs	
		%sp,%rs	General-purpose register (24 bits) → SP
		%sp,imm7	Immediate → SP
Integer arithmetic operation	add	%rd,%rs	16-bit addition between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	add/c		
	add/nc		
	add	%rd,imm7	16-bit addition of general-purpose register and immediate
	add.a	%rd,%rs	24-bit addition between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	add.a/c		
	add.a/nc		
	add.a	%sp,%rs	24-bit addition of SP and general-purpose register
		%rd,imm7	24-bit addition of general-purpose register and immediate
		%sp,imm7	24-bit addition of SP and immediate
	adc	%rd,%rs	16-bit addition with carry between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	adc/c		
	adc/nc		
	adc	%rd,imm7	16-bit addition of general-purpose register and immediate with carry
	sub	%rd,%rs	16-bit subtraction between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	sub/c		
	sub/nc		
	sub	%rd,imm7	16-bit subtraction of general-purpose register and immediate
	sub.a	%rd,%rs	24-bit subtraction between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	sub.a/c		
	sub.a/nc		
	sub.a	%sp,%rs	24-bit subtraction of SP and general-purpose register
		%rd,imm7	24-bit subtraction of general-purpose register and immediate
		%sp,imm7	24-bit subtraction of SP and immediate
	sbc	%rd,%rs	16-bit subtraction with carry between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	sbc/c		
	sbc/nc		
	sbc	%rd,imm7	16-bit subtraction of general-purpose register and immediate with carry
	cmp	%rd,%rs	16-bit comparison between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	cmp/c		
	cmp/nc		
	cmp	%rd,sign7	16-bit comparison of general-purpose register and immediate
	cmp.a	%rd,%rs	24-bit comparison between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	cmp.a/c		
	cmp.a/nc		
	cmp.a	%rd,imm7	24-bit comparison of general-purpose register and immediate
	cmc	%rd,%rs	16-bit comparison with carry between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	cmc/c		
	cmc/nc		
	cmc	%rd,sign7	16-bit comparison of general-purpose register and immediate with carry
Logical operation	and	%rd,%rs	Logical AND between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	and/c		
	and/nc		
	and	%rd,sign7	Logical AND of general-purpose register and immediate
	or	%rd,%rs	Logical OR between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	or/c		
	or/nc		
	or	%rd,sign7	Logical OR of general-purpose register and immediate
	xor	%rd,%rs	Exclusive OR between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	xor/c		
	xor/nc		
	xor	%rd,sign7	Exclusive OR of general-purpose register and immediate
	not	%rd,%rs	Logical inversion between general-purpose registers (1's complement) Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	not/c		
	not/nc		
	not	%rd,sign7	Logical inversion of general-purpose register and immediate (1's complement)

Classification	Mnemonic		Function
Shift and swap	sr	*%rd,%rs*	Logical shift to the right with the number of bits specified by the register
		%rd,imm7	Logical shift to the right with the number of bits specified by immediate
	sa	*%rd,%rs*	Arithmetic shift to the right with the number of bits specified by the register
		%rd,imm7	Arithmetic shift to the right with the number of bits specified by immediate
	sl	*%rd,%rs*	Logical shift to the left with the number of bits specified by the register
		%rd,imm7	Logical shift to the left with the number of bits specified by immediate
	swap	*%rd,%rs*	Bytewise swap on byte boundary in 16 bits
Immediate extension	ext	*imm13*	Extend operand in the following instruction
Conversion	cv.ab	*%rd,%rs*	Convert signed 8-bit data into 24 bits
	cv.as	*%rd,%rs*	Convert signed 16-bit data into 24 bits
	cv.al	*%rd,%rs*	Convert 32-bit data into 24 bits
	cv.la	*%rd,%rs*	Converts 24-bit data into 32 bits
	cv.ls	*%rd,%rs*	Converts 16-bit data into 32 bits
Branch	jpr	*sign10*	PC relative jump
	jpr.d	*%rb*	Delayed branching possible
	jpa	*imm7*	Absolute jump
	ipa.d	*%rb*	Delayed branching possible
	jrgt	*sign7*	PC relative conditional jump Branch condition: !Z & !(N ^ V)
	jrgt.d		Delayed branching possible
	jrge	*sign7*	PC relative conditional jump Branch condition: !(N ^ V)
	jrge.d		Delayed branching possible
	jrlt	*sign7*	PC relative conditional jump Branch condition: N ^ V
	jrlt.d		Delayed branching possible
	jrle	*sign7*	PC relative conditional jump Branch condition: Z | N ^ V
	jrle.d		Delayed branching possible
	jrugt	*sign7*	PC relative conditional jump Branch condition: !Z & !C
	jrugt.d		Delayed branching possible
	jruge	*sign7*	PC relative conditional jump Branch condition: !C
	jruge.d		Delayed branching possible
	jrult	*sign7*	PC relative conditional jump Branch condition: C
	jrult.d		Delayed branching possible
	jrule	*sign7*	PC relative conditional jump Branch condition: Z | C
	jrule.d		Delayed branching possible
	jreq	*sign7*	PC relative conditional jump Branch condition: Z
	jreq.d		Delayed branching possible
	jrne	*sign7*	PC relative conditional jump Branch condition: !Z
	jrne.d		Delayed branching possible
	call	*sign10*	PC relative subroutine call
	call.d	*%rb*	Delayed call possible
	calla	*imm7*	Absolute subroutine call
	calla.d	*%rb*	Delayed call possible
	ret		Return from subroutine
	ret.d		Delayed return possible
	int	*imm5*	Software interrupt
	intl	*imm5,imm3*	Software interrupt with interrupt level setting
	reti		Return from interrupt handling
	reti.d		Delayed call possible
	brk		Debug interrupt
	retd		Return from debug processing
System control	nop		No operation
	halt		HALT mode
	slp		SLEEP mode
	ei		Enable interrupts
	di		Disable interrupts

∗1 The `ld.a` instruction accesses memories in 32-bit length. During data transfer from a register to a memory, the 32-bit data in which the eight high-order bits are set to 0 is written to the memory. During reading from a memory, the eight high-order bits of the read data are ignored.

∗2 The S1C17701 does not include a coprocessor. Therefore, the coprocessor instructions are not available.

The symbols in the above table each have the meanings specified below.

Table 2.3.2 Symbol Meanings

Symbol	Description
%*rs*	General-purpose register, source
%*rd*	General-purpose register, destination
[%*rb*]	Memory addressed by general-purpose register
[%*rb*]+	Memory addressed by general-purpose register with address post-incremented
[%*rb*]-	Memory addressed by general-purpose register with address post-decremented
-[%*rb*]	Memory addressed by general-purpose register with address pre-decremented
%sp	Stack pointer
[%sp],[%sp+*imm7*]	Stack
[%sp]+	Stack with address post-incremented
[%sp]-	Stack with address post-decremented
-[%sp]	Stack with address pre-decremented
imm3,*imm5*,*imm7*,*imm13*	Unsigned immediate (numerals indicating bit length)
sign7,*sign10*	Signed immediate (numerals indicating bit length)

2.4 Vector Table

The vector table contains the vectors to the interrupt handler routines (handler routine start address) that will be read by the S1C17 Core to execute the handler when an interrupt occurs. The boot address from which the program starts running after a reset must be written to the top of the vector table.

The vector table is located at address 0x8000 in the S1C17701. The vector table base address can be read out from TTBR (Vector Table Base Register) located at address 0xffff80.

Table 2.4.1 shows the vector table of the S1C17701.

Table 2.4.1 Vector Table

Vector No. Software interrupt No.	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
0 (0x00)	0x8000	Reset	• Low input to the #RESET pin • Watchdog timer overflow [2]	1
1 (0x01)	0x8004	Address misaligned interrupt	Memory access instruction	2
–	(0xfffc00)	Debugging interrupt	`brk` instruction, etc.	3
2 (0x02)	0x8008	NMI	Watchdog timer overflow [2]	4
3 (0x03)	0x800c	reserved	–	–
4 (0x04)	0x8010	P0 port interrupt	P00–P07 port inputs	High [1]
5 (0x05)	0x8014	P1 port interrupt	P10–P17 port inputs	↑
6 (0x06)	0x8018	Stopwatch timer interrupt	• 100 Hz timer signal • 10 Hz timer signal • 1 Hz timer signal	
7 (0x07)	0x801c	Clock timer interrupt	• 32 Hz timer signal • 8 Hz timer signal • 2 Hz timer signal • 1 Hz timer signal	
8 (0x08)	0x8020	8-bit OSC1 timer interrupt	Compare match	
9 (0x09)	0x8024	SVD interrupt	Low supply voltage detected	
10 (0x0a)	0x8028	LCD interrupt	Frame signal	
11 (0x0b)	0x802c	PWM & capture timer interrupt	• Compare match A • Compare match B	
12 (0x0c)	0x8030	8-bit timer interrupt	Timer underflow	
13 (0x0d)	0x8034	16-bit timer Ch. 0 interrupt	Timer underflow	
14 (0x0e)	0x8038	16-bit timer Ch. 1 interrupt	Timer underflow	
15 (0x0f)	0x803c	16-bit timer Ch. 2 interrupt	Timer underflow	
16 (0x10)	0x8040	UART interrupt	• Transmit buffer empty • Receive buffer full • Receive error	
17 (0x11)	0x8044	Remote controller interrupt	• Data length counter underflow • Input rising edge detected • Input falling edge detected	
18 (0x12)	0x8048	SPI interrupt	• Transmit buffer empty • Receive buffer full	
19 (0x13)	0x804c	I^{2}C interrupt	• Transmit buffer empty • Receive buffer full	
20 (0x14) : 31 (0x1f)	0x8050 : 0x807c	reserved : reserved	– : –	↓ Low [1]

[1] When the same interrupt level is set

[2] Either reset or NMI can be selected as the watchdog timer interrupt with software.

0xffff80: Vector Table Base Register (TTBR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Vector Table Base Register (TTBR)	**0xffff80** (32 bits)	D31–24	**–**	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	**TTBR[23:0]**	Vector table base address	0x8000	0x8000	R	

2.5 Processor Information

The S1C17701 has the Processor ID Register (0xffff84) shown below that allow the application software to identify CPU core type.

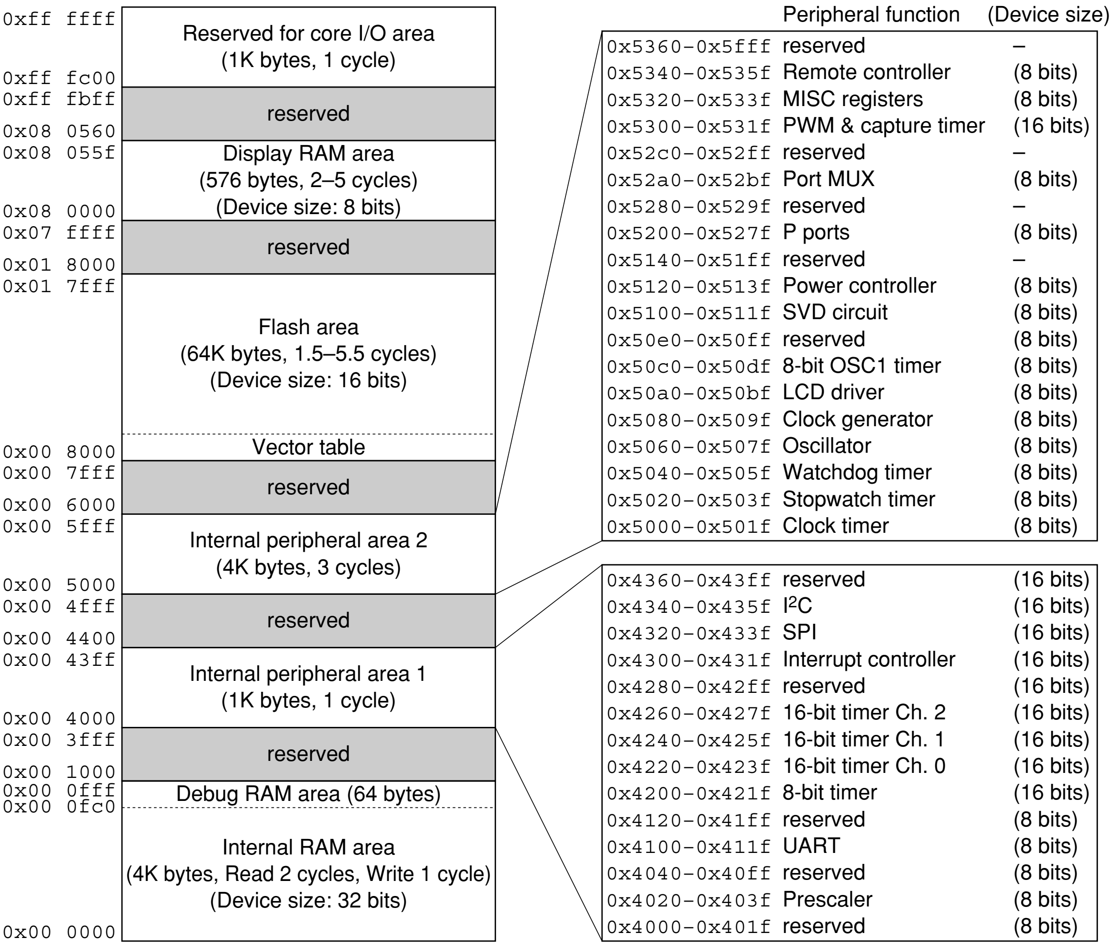
0xffff84: Processor ID Register (IDIR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Processor ID Register (IDIR)	**0xffff84** (8 bits)	D7–0	**IDIR[7:0]**	Processor ID 0x10: S1C17 Core	0x10	0x10	R	

This is a read-only register that contains the ID code to represent a processor model. The S1C17 Core’s ID code is 0x10.

3 Memory Map, Bus Control

Figure 3.1 shows the S1C17701 memory map.



0xff ffff	0xff fc00	Reserved for core I/O area (1K bytes, 1 cycle)
0xff fbff	0x08 0560	reserved
0x08 055f	0x08 0000	Display RAM area (576 bytes, 2–5 cycles) (Device size: 8 bits)
0x07 ffff	0x01 8000	reserved
0x01 7fff		Flash area (64K bytes, 1.5–5.5 cycles) (Device size: 16 bits)
	0x00 8000	Vector table
0x00 7fff	0x00 6000	reserved
0x00 5fff	0x00 5000	Internal peripheral area 2 (4K bytes, 3 cycles)
0x00 4fff	0x00 4400	reserved
0x00 43ff	0x00 4000	Internal peripheral area 1 (1K bytes, 1 cycle)
0x00 3fff	0x00 1000	reserved
0x00 0fff		Debug RAM area (64 bytes)
0x00 0fc0	0x00 0000	Internal RAM area (4K bytes, Read 2 cycles, Write 1 cycle) (Device size: 32 bits)

	Peripheral function	(Device size)
0x5360–0x5fff	reserved	–
0x5340–0x535f	Remote controller	(8 bits)
0x5320–0x533f	MISC registers	(8 bits)
0x5300–0x531f	PWM & capture timer	(16 bits)
0x52c0–0x52ff	reserved	–
0x52a0–0x52bf	Port MUX	(8 bits)
0x5280–0x529f	reserved	–
0x5200–0x527f	P ports	(8 bits)
0x5140–0x51ff	reserved	–
0x5120–0x513f	Power controller	(8 bits)
0x5100–0x511f	SVD circuit	(8 bits)
0x50e0–0x50ff	reserved	(8 bits)
0x50c0–0x50df	8-bit OSC1 timer	(8 bits)
0x50a0–0x50bf	LCD driver	(8 bits)
0x5080–0x509f	Clock generator	(8 bits)
0x5060–0x507f	Oscillator	(8 bits)
0x5040–0x505f	Watchdog timer	(8 bits)
0x5020–0x503f	Stopwatch timer	(8 bits)
0x5000–0x501f	Clock timer	(8 bits)

0x4360–0x43ff	reserved	(16 bits)
0x4340–0x435f	I^{2}C	(16 bits)
0x4320–0x433f	SPI	(16 bits)
0x4300–0x431f	Interrupt controller	(16 bits)
0x4280–0x42ff	reserved	(16 bits)
0x4260–0x427f	16-bit timer Ch. 2	(16 bits)
0x4240–0x425f	16-bit timer Ch. 1	(16 bits)
0x4220–0x423f	16-bit timer Ch. 0	(16 bits)
0x4200–0x421f	8-bit timer	(16 bits)
0x4120–0x41ff	reserved	(8 bits)
0x4100–0x411f	UART	(8 bits)
0x4040–0x40ff	reserved	(8 bits)
0x4020–0x403f	Prescaler	(8 bits)
0x4000–0x401f	reserved	(8 bits)

Figure 3.1 S1C17701 Memory Map

3.1 Bus Cycle

The CPU operates with CCLK as the operating clock. For CCLK, see Section 8.2, “Controlling the CPU Core Clock (CCLK).”

The period between a CCLK rising edge and the next rising edge is assumed to be one CCLK (= one bus cycle). As shown in Figure 3.1, the number of cycles required for one bus access depends on the peripheral or memory module. Furthermore, the number of bus accesses depends on the CPU instruction (access size) and device size.

Table 3.1.1 Number of Bus Accesses

Device size	CPU access size	Number of bus accesses
8 bits	8 bits	1
	16 bits	2
	32 bits*	4
16 bits	8 bits	1
	16 bits	1
	32 bits*	2
32 bits	8 bits	1
	16 bits	1
	32 bits*	1

* Handling the eight high-order bits during 32-bit accesses

During writing, the eight high-order bits are written as 0. During reading from a memory, the eight high-order bits are ignored. However, the stack operation in an interrupt handling reads/writes 32-bit data that consists of the PSR value as the high-order 8 bits and the return address as the low order 24 bits.

Number of bus cycles calculation example

Number of bus cycles when the CPU accesses the display RAM area (eight-bit device, set to two access cycles) by a 16-bit read or write instruction.

2 [cycles] × 2 [bus accesses] = 4 [CCLK cycles]

3.1.1 Restrictions on Access Size

The modules shown below have a restriction on the access size. Appropriate instructions should be used in programming.

Flash memory

The Flash memory allows only 16-bit write instructions for programming. Reading data from the Flash memory has no such restriction.

SPI, I^{2}C

The SPI and I^{2}C registers allow only 16-bit read/write instructions for accessing.

Other modules can be accessed with an 8-bit, 16-bit, or 32-bit instruction. However, reading for an unnecessary register may change the peripheral module status and it may cause a problem. Therefore, use the appropriate instructions according to the device size.

3.1.2 Restrictions on Instruction Execution Cycles

An instruction fetch and a data access are not performed simultaneously under one of the conditions listed below. This prolongs the instruction fetch cycle for the number of data area access cycles.

- When the S1C17701 executes the instruction stored in the Flash area and accesses data in the Flash area, display RAM area or internal peripheral area 2 (0x5000–)
- When the S1C17701 executes the instruction stored in the internal RAM area and accesses data in the internal RAM area

3.2 Flash Area

3.2.1 Internal Flash Memory

The 64K-byte area from address 0x8000 to address 0x17fff contains a Flash memory (4K bytes × 16 sectors) for storing application programs and data. Address 0x8000 is defined as the vector table base address, therefore a vector table (see Section 2.4, "Vector Table") must be placed from the beginning of the area. The Flash memory can be read in 1.5 to 5.5 cycles.

3.2.2 Flash Programming

The S1C17701 supports on-board programming of the Flash memory, it makes it possible to program the Flash memory with the application programs/data by using the debugger through the ICD Mini. Furthermore, the S1C17701 supports self-programming by the application program stored in the Flash memory. The Flash memory can be programmed in 16-bit units. For programming of the Flash memory, see Appendix B, "Flash Programming."
The Flash memory supports two erase methods, chip erase and sector erase. The table below lists the correspondence between addresses and sectors required for sector erase.

Note: The debugger supports chip erase only and does not allow erasing in sector units.

Table 3.2.2.1 Correspondence Between Memory Address and Flash Sector

S1C17701 address	Flash sector number	S1C17701 address	Flash sector number
0x0f000–0x0ffff	7	0x17000–0x17fff	15
0x0e000–0x0efff	6	0x16000–0x16fff	14
0x0d000–0x0dfff	5	0x15000–0x15fff	13
0x0c000–0x0cfff	4	0x14000–0x14fff	12
0x0b000–0x0bfff	3	0x13000–0x13fff	11
0x0a000–0x0afff	2	0x12000–0x12fff	10
0x09000–0x09fff	1	0x11000–0x11fff	9
0x08000–0x08fff	0	0x10000–0x10fff	8

Note: The 32 bits (0x17ffc–0x17fff) at the end of Sector 15 are reserved for the system as the protect bits. Do not program this area with data other than protect settings.

3.2.3 Protect Bits

In order to protect the memory contents, the Flash memory provides two protection features, write protection and data read protection, that can be configured for every 16K-byte areas. The write protection disables writing data to the configured area. The data-read protection disables reading data from the configured area (the read value is always 0x0000). However, it does not disable the instruction fetch operation by the CPU.
The Flash memory provides the protect bits listed below. Program the protect bit corresponding to the area to be protected to 0.

0x17ffc–0x17ffe: Flash Protect Bits

Address	Bit	Function	Setting				Init.	R/W	Remarks
0x17ffc (16 bits)	D15–4	reserved	–				–	–	
	D3	Flash write-protect bit for 0x14000–0x17fff	1	Writable	0	Protected	1	R/W	
	D2	Flash write-protect bit for 0x10000–0x13fff	1	Writable	0	Protected	1	R/W	
	D1	Flash write-protect bit for 0x0c000–0x0ffff	1	Writable	0	Protected	1	R/W	
	D0	Flash write-protect bit for 0x08000–0x0bfff	1	Writable	0	Protected	1	R/W	
0x17ffe (16 bits)	D15–4	reserved	–				–	–	
	D3	Flash data-read-protect bit for 0x14000–0x17fff	1	Readable	0	Protected	1	R/W	
	D2	Flash data-read-protect bit for 0x10000–0x13fff	1	Readable	0	Protected	1	R/W	
	D1	Flash data-read-protect bit for 0x0c000–0x0ffff	1	Readable	0	Protected	1	R/W	
	D0	reserved	1				1	R/W	Always set to 1.

Notes:
- Be sure not to locate the area with data-read protection into the .data and .rodata sections.
- Be sure to set D0 of address 0x17ffe to 1. If it is set to 0, the program cannot be booted.

3.2.4 Access Control for the Flash Controller

The S1C17701 on-chip Flash memory is accessed via the exclusive Flash controller. A MISC register is used to set the access condition for the Flash controller.

Setting number of read access cycles for the Flash controller

In order to read data from the Flash memory properly, set the appropriate number of read access cycles according to the CCLK frequency using the FLCYC[2:0] bits (D[2:0]/MISC_FL register).

0x5320: FLASHC Control Register (MISC_FL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
FLASHC Control Register (MISC_FL)	**0x5320** (8 bits)	D7–3	**–**	reserved	–		–	–	0 when being read.
		D2–0	**FLCYC[2:0]**	FLASHC read access cycle	FLCYC[2:0]	Read cycle	0x3	R/W	
					0x7–0x5	reserved			
					0x4	1.5 cycles			
					0x3	5.5 cycles			
					0x2	4.5 cycles			
					0x1	3.5 cycles			
					0x0	2.5 cycles			

D[7:3] Reserved

D[2:0] FLCYC[2:0]: FLASHC Read Access Cycle Setup Bits

Sets the number of read access cycles for the Flash controller.

Table 3.2.4.1 Setting Read Access Cycles for the Flash Controller

FLCYC[2:0]	Number of read access cycles	CCLK frequency
0x7–0x5	Reserved	–
0x4	1.5 cycles	3.3 MHz max.
0x3	5.5 cycles	8.2 MHz max.
0x2	4.5 cycles	8.2 MHz max.
0x1	3.5 cycles	8.2 MHz max.
0x0	2.5 cycles	6 MHz max.

(Default: 0x3)

Note: Be sure to avoid setting a number of read access cycles that exceeds the maximum allowable CCLK frequency, as it may cause a malfunction.

3.3 Internal RAM Area

3.3.1 Internal RAM

The S1C17701 contains a RAM in the 4K-byte area from address 0x0 to address 0xfff. The RAM is accessed in two cycles for reading or one cycle for writing and allows high-speed execution of the instruction codes copied into it as well as storing variables and other data.

Note: The 64-byte area at the end of the RAM (0xfc0–0xfff) is reserved for the on-chip debugger. When using the debug functions under application development, do not access this area from the application program.
This area can be used for applications of mass-produced devices that do not need debugging.

3.4 Display RAM Area

3.4.1 Display RAM

The display RAM for the on-chip LCD driver is located in the 576-byte area from address 0x80000 to address 0x8055f. The display RAM is accessed in two to five cycles as an eight-bit device. It can be used as a general-purpose RAM when it is not used for display. See Section 22.5, "Display Memory," for details of the display memory.

3.4.2 Access Control for the SRAM Controller

The S1C17701 display RAM is accessed via the exclusive SRAM controller. A MISC register is used to set the access condition for the SRAM controller.

Setting number of access cycles for the SRAM controller

In order to read/write data from/to the display RAM properly, set the appropriate number of access cycles according to the CCLK frequency using the SRCYC[1:0] bits (D[1:0]/ MISC_SR register).

0x5321: SRAMC Control Register (MISC_SR)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
SRAMC Control Register (MISC_SR)	**0x5321** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**SRCYC[1:0]**	SRAMC access cycle	SRCYC[1:0]	Access cycle	0x3	R/W	
					0x3	5 cycles			
					0x2	4 cycles			
					0x1	3 cycles			
					0x0	2 cycles			

D[7:2] Reserved

D[1:0] SRCYC[1:0]: SRAMC Access Cycle Setup Bits

Sets the number of SRAM (display RAM) controller access cycle.

Table 3.4.2.1 Setting Access Cycles for the SRAM Controller

SRCYC[1:0]	Number of access cycles	CCLK frequency
0x3	5 cycles	8.2 MHz max.
0x2	4 cycles	8.2 MHz max.
0x1	3 cycles	8.2 MHz max.
0x0	2 cycles	6 MHz max.

(Default: 0x3)

3.5 Internal Peripheral Area

The I/O and control registers for the internal peripheral modules are located in the 1K-byte area beginning with address 0x4000 and the 4K-byte area beginning with address 0x5000.

3.5.1 Internal Peripheral Area 1 (0x4000–)

The internal peripheral area 1 beginning with address 0x4000 contains the I/O memory for the peripheral functions listed below and this area can be accessed in one cycle.

- Prescaler (PSC, 8-bit device)
- UART (UART, 8-bit device)
- 8-bit timer (T8F, 16-bit device)
- 16-bit timers (T16, 16-bit device)
- Interrupt controller (ITC, 16-bit device)
- SPI (SPI, 16-bit device)
- I^{2}C (I2C, 16-bit device)

3.5.2 Internal Peripheral Area 2 (0x5000–)

The internal peripheral area 2 beginning with address 0x5000 contains the I/O memory for the peripheral functions listed below and this area can be accessed in three cycles.

- Clock timer (CT, 8-bit device)
- Stopwatch timer (SWT, 8-bit device)
- Watchdog timer (WDT, 8-bit device)
- Oscillator (OSC, 8-bit device)
- Clock generator (CLG, 8-bit device)
- LCD driver (LCD, 8-bit device)
- 8-bit OSC1 timer (T8OSC1, 8-bit device)
- SVD circuit (SVD, 8-bit device)
- Power supply circuit (VD1, 8-bit device)
- I/O port & port MUX (P, 8-bit device)
- PWM & capture timer (T16E, 16-bit device)
- MISC register (MISC, 8-bit device)
- Remote controller (REMC, 8-bit device)

3.5.3 I/O Map

This section shows the I/O map table for the internal peripheral area. For details of each control register, see the I/O register list in Appendix or description for each peripheral module.

Table 3.5.3.1 I/O Map (Internal Peripheral Area 1)

Peripheral	Address	Register name		Function
Prescaler (8-bit device)	0x4020	PSC_CTL	Prescaler Control Register	Starts/stops the prescaler.
	0x4021–0x403f	–	–	Reserved
UART (with IrDA) (8-bit device)	0x4100	UART_ST	UART Status Register	Indicates transfer, buffer and error statuses.
	0x4101	UART_TXD	UART Transmit Data Register	Transmit data
	0x4102	UART_RXD	UART Receive Data Register	Receive data
	0x4103	UART_MOD	UART Mode Register	Sets transfer data format.
	0x4104	UART_CTL	UART Control Register	Controls data transfer.
	0x4105	UART_EXP	UART Expansion Register	Sets IrDA mode.
	0x4106–0x411f	–	–	Reserved
8-bit timer (with fine mode) (16-bit device)	0x4200	T8F_CLK	8-bit Timer Input Clock Select Register	Selects a prescaler output clock.
	0x4202	T8F_TR	8-bit Timer Reload Data Register	Sets reload data.
	0x4204	T8F_TC	8-bit Timer Counter Data Register	Counter data
	0x4206	T8F_CTL	8-bit Timer Control Register	Sets the timer mode and starts/stops the timer.
	0x4208–0x421f	–	–	Reserved
16-bit timer Ch. 0 (16-bit device)	0x4220	T16_CLK0	16-bit Timer Ch.0 Input Clock Select Register	Selects a prescaler output clock.
	0x4222	T16_TR0	16-bit Timer Ch.0 Reload Data Register	Sets reload data.
	0x4224	T16_TC0	16-bit Timer Ch.0 Counter Data Register	Counter data
	0x4226	T16_CTL0	16-bit Timer Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
	0x4228–0x423f	–	–	Reserved
16-bit timer Ch. 1 (16-bit device)	0x4240	T16_CLK1	16-bit Timer Ch.1 Input Clock Select Register	Selects a prescaler output clock.
	0x4242	T16_TR1	16-bit Timer Ch.1 Reload Data Register	Sets reload data.
	0x4244	T16_TC1	16-bit Timer Ch.1 Counter Data Register	Counter data
	0x4246	T16_CTL1	16-bit Timer Ch.1 Control Register	Sets the timer mode and starts/stops the timer.
	0x4248–0x425f	–	–	Reserved
16-bit timer Ch. 2 (16-bit device)	0x4260	T16_CLK2	16-bit Timer Ch.2 Input Clock Select Register	Selects a prescaler output clock.
	0x4262	T16_TR2	16-bit Timer Ch.2 Reload Data Register	Sets reload data.
	0x4264	T16_TC2	16-bit Timer Ch.2 Counter Data Register	Counter data
	0x4266	T16_CTL2	16-bit Timer Ch.2 Control Register	Sets the timer mode and starts/stops the timer.
	0x4268–0x427f	–	–	Reserved
Interrupt controller (16-bit device)	0x4300	ITC_IFLG	Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x4302	ITC_EN	Interrupt Enable Register	Enables/disables each maskable interrupt.
	0x4304	ITC_CTL	ITC Control Register	Enables/disables the ITC.
	0x4306	ITC_ELV0	External Interrupt Level Setup Register 0	Sets the P0 and P1 interrupt levels and trigger modes.
	0x4308	ITC_ELV1	External Interrupt Level Setup Register 1	Sets the stopwatch timer and clock timer interrupt levels and trigger modes.
	0x430a	ITC_ELV2	External Interrupt Level Setup Register 2	Sets the 8-bit OSC1 timer and SVD interrupt levels and trigger modes.
	0x430c	ITC_ELV3	External Interrupt Level Setup Register 3	Sets the LCD and PWM & capture timer interrupt levels and trigger modes.
	0x430e	ITC_ILV0	Internal Interrupt Level Setup Register 0	Sets the 8-bit timer and 16-bit timer Ch. 0 interrupt levels.
	0x4310	ITC_ILV1	Internal Interrupt Level Setup Register 1	Sets the 16-bit timer Ch. 1 and 16-bit timer Ch. 2 interrupt levels.
	0x4312	ITC_ILV2	Internal Interrupt Level Setup Register 2	Sets the UART and remote controller interrupt levels.
	0x4314	ITC_ILV3	Internal Interrupt Level Setup Register 3	Sets the SPI and I^{2}C interrupt levels.
	0x4316–0x431f	–	–	Reserved
SPI (16-bit device)	0x4320	SPI_ST	SPI Status Register	Indicates transfer and buffer statuses.
	0x4322	SPI_TXD	SPI Transmit Data Register	Transmit data
	0x4324	SPI_RXD	SPI Receive Data Register	Receive data
	0x4326	SPI_CTL	SPI Control Register	Sets the SPI mode and enables data transfer.
	0x4328–0x433f	–	–	Reserved
I^{2}C (16-bit device)	0x4340	I2C_EN	I^{2}C Enable Register	Enables the I^{2}C module.
	0x4342	I2C_CTL	I^{2}C Control Register	Controls the I^{2}C operation and indicates transfer status.
	0x4344	I2C_DAT	I^{2}C Data Register	Transmit/receive data
	0x4346	I2C_ICTL	I^{2}C Interrupt Control Register	Controls the I^{2}C interrupt.
	0x4348–0x435f	–	–	Reserved

Table 3.5.3.2 I/O Map (Internal Peripheral Area 2)

Peripheral	Address	Register name		Function
Clock timer (8-bit device)	0x5000	CT_CTL	Clock Timer Control Register	Resets and starts/stops the timer.
	0x5001	CT_CNT	Clock Timer Counter Register	Counter data
	0x5002	CT_IMSK	Clock Timer Interrupt Mask Register	Enables/disables interrupt.
	0x5003	CT_IFLG	Clock Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x5004–0x501f	–	–	Reserved
Stopwatch timer (8-bit device)	0x5020	SWT_CTL	Stopwatch Timer Control Register	Resets and starts/stops the timer.
	0x5021	SWT_BCNT	Stopwatch Timer BCD Counter Register	BCD counter data
	0x5022	SWT_IMSK	Stopwatch Timer Interrupt Mask Register	Enables/disables interrupt.
	0x5023	SWT_IFLG	Stopwatch Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x5024–0x503f	–	–	Reserved
Watchdog timer (8-bit device)	0x5040	WDT_CTL	Watchdog Timer Control Register	Resets and starts/stops the timer.
	0x5041	WDT_ST	Watchdog Timer Status Register	Sets the timer mode and indicates NMI status.
	0x5042–0x505f	–	–	Reserved
Oscillator (8-bit device)	0x5060	OSC_SRC	Clock Source Select Register	Selects a clock source.
	0x5061	OSC_CTL	Oscillation Control Register	Controls oscillation.
	0x5062	OSC_NFEN	Noise Filter Enable Register	Enables/disables noise filters.
	0x5063	OSC_LCLK	LCD Clock Setup Register	Sets up the LCD clock.
	0x5064	OSC_FOUT	FOUT Control Register	Controls clock output.
	0x5065	OSC_T8OSC1	T8OSC1 Clock Control Register	Sets up the 8-bit OSC1 timer clock.
	0x5066–0x507f	–	–	Reserved
Clock generator (8-bit device)	0x5080	CLG_PCLK	PCLK Control Register	Controls the PCLK output.
	0x5081	CLG_CCLK	CCLK Control Register	Configures the CCLK division ratio.
	0x5082–0x509f	–	–	Reserved
LCD driver (8-bit device)	0x50a0	LCD_DCTL	LCD Display Control Register	Controls the LCD display.
	0x50a1	LCD_CADJ	LCD Contrast Adjust Register	Controls the contrast.
	0x50a2	LCD_CCTL	LCD Clock Control Register	Controls the LCD clock duty.
	0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	Controls the LCD drive voltage regulator.
	0x50a4	LCD_PWR	LCD Power Voltage Booster Control Register	Controls the LCD voltage booster.
	0x50a5	LCD_IMSK	LCD Interrupt Mask Register	Enables/disables interrupt.
	0x50a6	LCD_IFLG	LCD Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x50a7–0x50bf	–	–	Reserved
8-bit OSC1 timer (8-bit device)	0x50c0	T8OSC1_CTL	8-bit OSC1 Timer Control Register	Sets the timer mode and starts/stops the timer.
	0x50c1	T8OSC1_CNT	8-bit OSC1 Timer Counter Data Register	Counter data
	0x50c2	T8OSC1_CMP	8-bit OSC1 Timer Compare Data Register	Sets compare data.
	0x50c3	T8OSC1_IMSK	8-bit OSC1 Timer Interrupt Mask Register	Enables/disables interrupt.
	0x50c4	T8OSC1_IFLG	8-bit OSC1 Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x50c5–0x50df	–	–	Reserved
SVD circuit (8-bit device)	0x5100	SVD_EN	SVD Enable Register	Enables/disables the SVD operation.
	0x5101	SVD_CMP	SVD Compare Voltage Register	Sets compare voltage.
	0x5102	SVD_RSLT	SVD Detection Result Register	Voltage detection results
	0x5103	SVD_IMSK	SVD Interrupt Mask Register	Enables/disables interrupt.
	0x5104	SVD_IFLG	SVD Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x5105–0x511f	–	–	Reserved
Power supply circuit (8-bit device)	0x5120	VD1_CTL	V_{D1} Control Register	Controls the V_{D1} voltage and heavy load protection mode.
	0x5121–0x513f	–	–	Reserved
P port & port MUX (8-bit device)	0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
	0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
	0x5202	P0_IO	P0 Port I/O Direction Control Register	Selects the P0 port I/O direction.
	0x5203	P0_PU	P0 Port Pull-up Control Register	Controls the P0 port pull-up resistor.
	0x5204	P0_SM	P0 Port Schmitt Trigger Control Register	Controls the P0 port Schmitt trigger input.
	0x5205	P0_IMSK	P0 Port Interrupt Mask Register	Enables/disables the P0 port interrupt.
	0x5206	P0_EDGE	P0 Port Interrupt Edge Select Register	Selects the signal edge for generating P0 port interrupts.
	0x5207	P0_IFLG	P0 Port Interrupt Flag Register	Indicates/resets the P0 port interrupt occurrence status.
	0x5208	P0_CHAT	P0 Port Chattering Filter Control Register	Controls the P0 port chattering filter.
	0x5209	P0_KRST	P0 Port Key-Entry Reset Configuration Register	Configures the P0 port key-entry reset function.
	0x520a–0x520f	–	–	Reserved
	0x5210	P1_IN	P1 Port Input Data Register	P1 port input data
	0x5211	P1_OUT	P1 Port Output Data Register	P1 port output data
	0x5212	P1_IO	P1 Port I/O Direction Control Register	Selects the P1 port I/O direction.
	0x5213	P1_PU	P1 Port Pull-up Control Register	Controls the P1 port pull-up resistor.
	0x5214	P1_SM	P1 Port Schmitt Trigger Control Register	Controls the P1 port Schmitt trigger input.
	0x5215	P1_IMSK	P1 Port Interrupt Mask Register	Enables/disables the P1 port interrupt.

Peripheral	Address	Register name		Function
P port & port MUX (8-bit device)	0x5216	P1_EDGE	P1 Port Interrupt Edge Select Register	Selects the signal edge for generating P1 port interrupts.
	0x5217	P1_IFLG	P1 Port Interrupt Flag Register	Indicates/resets the P1 port interrupt occurrence status.
	0x5218–0x521f	–	–	Reserved
	0x5220	P2_IN	P2 Port Input Data Register	P2 port input data
	0x5221	P2_OUT	P2 Port Output Data Register	P2 port output data
	0x5222	P2_IO	P2 Port I/O Direction Control Register	Selects the P2 port I/O direction.
	0x5223	P2_PU	P2 Port Pull-up Control Register	Controls the P2 port pull-up resistor.
	0x5224	P2_SM	P2 Port Schmitt Trigger Control Register	Controls the P2 port Schmitt trigger input.
	0x5225–0x522f	–	–	Reserved
	0x5230	P3_IN	P3 Port Input Data Register	P3 port input data
	0x5231	P3_OUT	P3 Port Output Data Register	P3 port output data
	0x5232	P3_IO	P3 Port I/O Direction Control Register	Selects the P3 port I/O direction.
	0x5233	P3_PU	P3 Port Pull-up Control Register	Controls the P3 port pull-up resistor.
	0x5234	P3_SM	P3 Port Schmitt Trigger Control Register	Controls the P3 port Schmitt trigger input.
	0x5235–0x527f	–	–	Reserved
	0x52a0	P0_PMUX	P0 Port Function Select Register	Selects the P0 port function.
	0x52a1	P1_PMUX	P1 Port Function Select Register	Selects the P1 port function.
	0x52a2	P2_PMUX	P2 Port Function Select Register	Selects the P2 port function.
	0x52a3	P3_PMUX	P3 Port Function Select Register	Selects the P3 port function.
	0x52a4–0x52bf	–	–	Reserved
PWM & capture timer (16-bit device)	0x5300	T16E_CA	PWM Timer Compare Data A Register	Sets compare data A.
	0x5302	T16E_CB	PWM Timer Compare Data B Register	Sets compare data B.
	0x5304	T16E_TC	PWM Timer Counter Data Register	Counter data
	0x5306	T16E_CTL	PWM Timer Control Register	Sets the timer mode and starts/stops the timer.
	0x5308	T16E_CLK	PWM Timer Input Clock Select Register	Selects a prescaler output clock.
	0x530a	T16E_IMSK	PWM Timer Interrupt Mask Register	Enables/disables interrupt.
	0x530c	T16E_IFLG	PWM Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x530e–0x531f	–	–	Reserved
MISC register (8-bit device)	0x5320	MISC_FL	FLASHC Control Register	Sets FLASHC access condition.
	0x5321	MISC_SR	SRAMC Control Register	Sets SRAMC access condition.
	0x5322	MISC_OSC1	OSC1 Peripheral Control Register	Selects the OSC1 peripheral operation in debug mode.
	0x5323–0x533f	–	–	Reserved
Remote controller (8-bit device)	0x5340	REMC_CFG	REMC Configuration Register	Selects/enables transmission/reception
	0x5341	REMC_PSC	REMC Prescaler Clock Select Register	Selects a prescaler output clock.
	0x5342	REMC_CARH	REMC H Carrier Length Setup Register	Sets up the H period of the carrier.
	0x5343	REMC_CARL	REMC L Carrier Length Setup Register	Sets up the L period of the carrier.
	0x5344	REMC_ST	REMC Status Register	Transmit/receive bit
	0x5345	REMC_LCNT	REMC Length Counter Register	Sets the transmit/receive data length.
	0x5346	REMC_IMSK	REMC Interrupt Mask Register	Enables/disables interrupt.
	0x5347	REMC_IFLG	REMC Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x5348–0x535f	–	–	Reserved

Note: Do not access the “Reserved” address in the table above and unused areas in the peripheral area that are not described in the table from the application program.

3.6 S1C17 Core I/O Area

The 1K-byte area from address 0xfffc00 to address 0xffffff is the I/O area for the CPU core in which the I/O registers listed in the table below are located.

Table 3.6.1 I/O Map (S1C17 Core I/O Area)

Peripheral	Address	Register name		Function
S1C17 Core I/O	0xffff80	TTBR	Vector Table Base Register	Indicates the vector table base address.
	0xffff84	IDIR	Processor ID Register	Indicates the processor ID.
	0xffff90	DBRAM	Debug RAM Base Register	Indicates the debug RAM base address.

See Section 2.4, “Vector Table,” and Section 2.5, “Processor Information,” for TTBR and IDIR, respectively. For DBRAM, see Chapter 24, “On-chip Debugger (DBG).”

THIS PAGE IS BLANK.

4 Power Supply

4.1 Power Supply Voltage

The operating voltage range of the S1C17701 is as follows:

For normal operation: 1.8 V to 3.6 V
For Flash programming: 2.7 V to 3.6 V

Supply a voltage within the range to the V_{DD} pins with the V_{SS} pins as the GND level. The S1C17701 provides two V_{DD} pins and three V_{SS} pins. Do not leave any pins open and be sure to connect them to + power source and GND.

4.2 Internal Power Supply Circuit

The S1C17701 has a built-in power supply circuit shown in Figure 4.2.1 to generate all the power voltages required for the internal circuits. The power supply module consists of three circuits.

Table 4.2.1 Power Supply Circuit

Circuit	Power supply circuit	Output voltage
Oscillator and internal logic circuits	Internal logic voltage regulator	V_{D1}
LCD system voltage regulator	Power voltage booster	V_{DD} or V_{D2}
LCD driver	LCD system voltage regulator	V_{C1} to V_{C5}

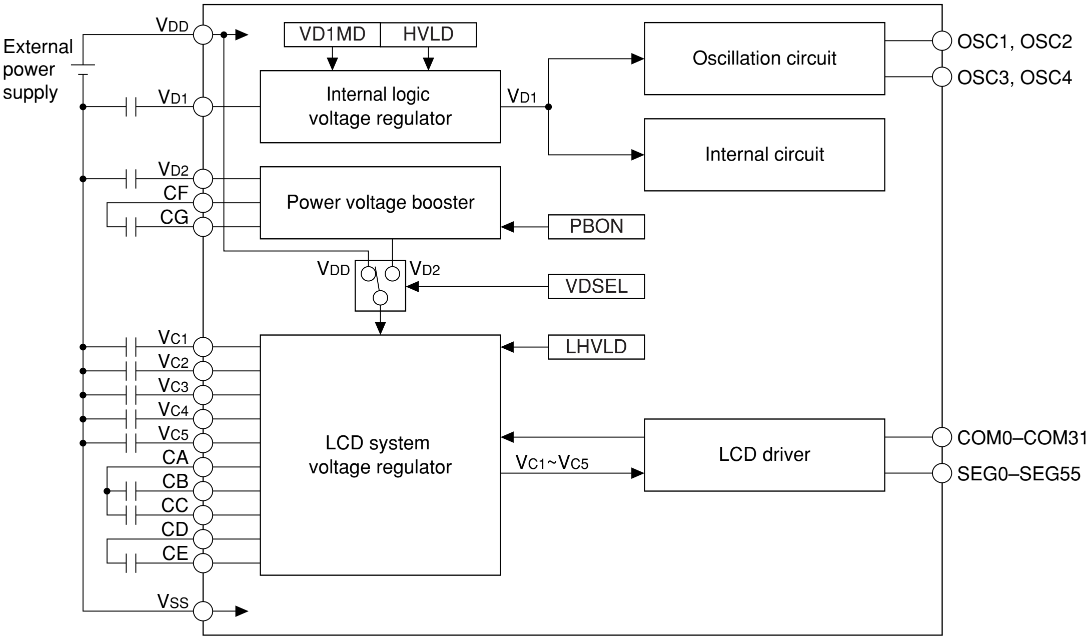


Figure 4.2.1 Configuration of Power Supply Circuit

Note: Be sure to avoid using the V_{D1}, V_{D2}, and V_{C1}–V_{C5} pin outputs to drive external circuits.

Internal logic voltage regulator

The internal logic voltage regulator generates the V_{D1} operating voltage for the internal logic circuits and oscillators. The V_{D1} voltage value can be switched in the program; set it to 1.8 V for normal operation and 2.5 V for Flash programming.

Power voltage booster

The power voltage booster generates the V_{D2} operating voltage for the LCD system voltage regulator. Either V_{DD} or V_{D2} can be selected as the power source for the LCD system voltage regulator according to the V_{DD} supply voltage value.

Table 4.2.2 Power Source for LCD System Voltage Regulator

Power supply voltage V_{DD}	Power source for the LCD system voltage regulator
1.8 to 2.5 V	V_{D2} ($\fallingdotseq V_{DD} \times 2$)
2.5 to 3.6 V	V_{DD}

LCD system voltage regulator

The LCD system voltage regulator generates the 1/5-bias LCD drive voltages V_{C1}, V_{C2}, V_{C3}, V_{C4}, and V_{C5}. In the S1C17701, the LCD drive voltage is supplied to the built-in LCD driver that drives the LCD panel connected to the SEG and COM pins.

Note: If V_{DD} is used as the power source for the LCD system voltage regulator when V_{DD} is 2.5 V or less, the V_{C1} to V_{C5} voltages cannot be generated within the specifications.

4.3 Controlling the Power Supply Circuit

In order to generate the internal operating voltage properly according to the power supply voltage and operating mode, or to reduce current consumption, the power supply circuit is designed to be controlled with software.

Switching the operating mode

The S1C17701 has two kinds of operating modes.

1. Normal operation mode
   This mode is provided for running the application program.
   V_{DD} = 1.8 to 3.6 V, internal operating voltage V_{D1} = 1.8 V

2. Flash erase/program mode
   This mode is provided for erasing and programming the Flash memory.
   V_{DD} = 2.7 to 3.6 V, internal operating voltage V_{D1} = 2.5 V

The V_{D1} voltage value must be switched according to the operating mode as shown above using the VD1MD bit (D0/VD1_CTL register). Normally set VD1MD to 0 (V_{D1} = 1.8 V, default setting). It should be set to 1 before erasing/programming the Flash memory.

∗ **VD1MD**: Flash Erase/Program Mode Bit in the V_{D1} Control (VD1_CTL) Register (D0/0x5120)

Note: When the operating mode is switched, the internal operating voltage requires 5 ms (max.) to stabilize. Flash memory programming should be started after the stabilization time has elapsed.

Controlling the LCD power source

The LCD system voltage regulator must be driven with a 2.5 V or more power voltage to generate appropriate LCD drive voltages V_{C1} to V_{C5}. When the power supply voltage (V_{DD}) is within the range from 1.8 V to 2.5 V, use the power voltage booster to generate double the V_{DD} voltage and drive the LCD system voltage regulator with the V_{D2} output voltage. Set the PBON bit (D0/LCD_PWR register) to 1 to turn the power voltage booster on. In addition, set the VDSEL bit (D1/LCD_PWR register) to 1 to drive the LCD system voltage regulator with the V_{D2} voltage output from the power voltage booster. PBON must be set to 1 before the drive voltage can be switched to V_{D2}.

∗ **PBON**: Power Voltage Booster Control Bit in the LCD Power Voltage Booster Control (LCD_PWR) Register (D0/0x50a4)
∗ **VDSEL**: Regulator Power Source Select Bit in the LCD Power Voltage Booster Control (LCD_PWR) Register (D1/0x50a4)

When the power supply voltage (V_{DD}) is 2.5 V or more, drive the LCD system voltage regulator with V_{DD}. The power voltage booster should be turned off to reduce current consumption. In this case, PBON and VDSEL are both set to 0 (default).

Note: When the power voltage booster is turned on, the V_{D2} output voltage requires about 1 ms to stabilize. Do not switch the power source for the LCD system voltage regulator to V_{D2} until the stabilization time has elapsed.

The LCD drive voltages V_{C1} to V_{C5} will be supplied to the LCD driver by setting the DSPC[1:0] bits (D[1:0]/LCD_DCTL register) to a value other than 0x0 (display off).

∗ **DSPC[1:0]**: LCD Display Control Bits in the LCD Display Control (LCD_DCTL) Register (D[1:0]/0x50a0)

When the internal LCD driver is not used, the power voltage booster and LCD system voltage regulator should be turned off to reduce current consumption. Set PBON, VDSEL, and DSPC[1:0] to 0 (default).

Power control bit settings

Table 4.3.1 lists the power control bit settings in different operating conditions.

Table 4.3.1 Power Control Bit Settings

Condition			Control bits			
Operating mode	**V_{DD}**	**LCD driver**	**VD1MD**	**PBON**	**VDSEL**	**DSPC[1:0]**
Normal operation	1.8 to 2.5 V	Used	0	1	1	Other than 0x0
	2.5 to 3.6 V	Used	0	0	0	Other than 0x0
	1.8 to 3.6 V	Not used	0	0	0	0x0
Flash erase/ program	1.8 to 2.7 V	–	(use prohibited)			
	2.7 to 3.6 V	Used	1	0	0	Other than 0x0
	2.7 to 3.6 V	Not used	1	0	0	0x0

For the DSPC[1:0] settings, see “0x50a0: LCD Display Control Register (LCD_DCTL)” in Section 22.8.

4.4 Heavy Load Protection Function

In order to ensure a stable circuit behavior and LCD display quality even if the power supply voltage fluctuates due to driving an external load, the internal logic voltage regulator and the LCD system voltage regulator have a heavy load protection function.
The internal logic voltage regulator enters heavy load protection mode by writing 1 to the HVLD bit (D4/VD1_CTL register) and it ensures stable V_{D1} output. Use the heavy load protection function when a heavy load such as a lamp or buzzer is driven with a port output.

∗ **HVLD**: V_{D1} Heavy Load Protection Mode Bit in the V_{D1} Control (VD1_CTL) Register (D4/0x5120)

The LCD system voltage regulator enters heavy load protection mode by writing 1 to the LHVLD bit (D4/ LCD_VREG register) and it ensures stable V_{C1}–V_{C5} outputs. Use the heavy load protection function when the LCD display has inconsistencies in density.

∗ **LHVLD**: LCD Heavy Load Protection Mode Bit in the LCD Voltage Regulator Control (LCD_VREG) Register (D4/0x50a3)

Note: Current consumption increases in heavy load protection mode, therefore do not set heavy load protection mode with software if unnecessary.

4.5 Details of Control Registers

Table 4.5.1 List of Power Control Registers

Address	Register name		Function
0x5120	VD1_CTL	V_{D1} Control Register	Controls the V_{D1} voltage and heavy load protection mode.
0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	Controls the LCD drive voltage regulator.
0x50a4	LCD_PWR	LCD Power Voltage Booster Control Register	Controls the LCD voltage booster.

The following describes each power control register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5120: V_{D1} Control Register (VD1_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
V_{D1} Control Register (VD1_CTL)	0x5120 (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	HVLD	V_{D1} heavy load protection mode	1	On	0	Off	0	R/W	
		D3–1	–	reserved	–				–	–	0 when being read.
		D0	VD1MD	Flash erase/program mode	1	Flash (2.5 V)	0	Norm.(1.8 V)	0	R/W	

D[7:5] Reserved

D4 HVLD: V_{D1} Heavy Load Protection Mode Bit

Sets the internal logic voltage regulator into heavy load protection mode.

1 (R/W): Heavy load protection On

0 (R/W): Heavy load protection Off (default)

The internal logic voltage regulator enters heavy load protection mode by writing 1 to HVLD and it ensures stable V_{D1} output. Use the heavy load protection function when a heavy load such as a lamp or buzzer is driven with a port output. Current consumption increases in heavy load protection mode, therefore do not set if unnecessary.

D[3:1] Reserved

D0 VD1MD: Flash Erase/Program Mode Bit

Selects the V_{D1} internal operating voltage value (operating mode).

1 (R/W): V_{D1} = 2.5 V, Flash erase/program mode

0 (R/W): V_{D1} = 1.8 V, Normal operation mode (default)

Normally set VD1MD to 0 (V_{D1} = 1.8 V, default setting). It should be set to 1 before erasing/programming the Flash memory.

Note: When the operating mode is switched, the internal operating voltage requires 5 ms (max.) to stabilize. Flash memory programming should be started after the stabilization time has elapsed.

0x50a3: LCD Voltage Regulator Control Register (LCD_VREG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
LCD Voltage Regulator Control Register (LCD_VREG)	**0x50a3** (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	**LHVLD**	LCD heavy load protection mode	1	On	0	Off	0	R/W	
		D3–0	–	reserved	–				–	–	0 when being read.

D[7:5] Reserved

D4 LHVLD: LCD Heavy Load Protection Mode Bit

Sets the LCD system voltage regulator into heavy load protection mode.

1 (R/W): Heavy load protection On

0 (R/W): Heavy load protection Off (default)

The LCD system voltage regulator enters heavy load protection mode by writing 1 to LHVLD and it ensures stable V_{C1}–V_{C5} outputs. Use the heavy load protection function when the LCD display has inconsistencies in density. Current consumption increases in heavy load protection mode, therefore do not set if unnecessary.

D[3:0] Reserved

0x50a4: LCD Power Voltage Booster Control Register (LCD_PWR)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
LCD Power Voltage Booster Control Register (LCD_PWR)	**0x50a4** (8 bits)	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**VDSEL**	Regulator power source select	1	V_{D2}	0	V_{DD}	0	R/W	
		D0	**PBON**	Power voltage booster control	1	On	0	Off	0	R/W	

D[7:2] Reserved

D1 VDSEL: Regulator Power Source Select Bit

Selects the power source voltage for the LCD system voltage regulator.

1 (R/W): V_{D2}

0 (R/W): V_{DD} (default)

When the power supply voltage (V_{DD}) is within the range from 1.8 V to 2.5 V, write 1 to VDSEL to drive the LCD system voltage regulator with the V_{D2} voltage output from the power voltage booster. Before this setting though, write 1 to PBON (D0) to turn the power voltage booster on.

When the power supply voltage (V_{DD}) is 2.5 V or more, write 0 to VDSEL to drive the LCD system voltage regulator with V_{DD}. In this case, the power voltage booster should be turned off to reduce current consumption.

D0 PBON: Power Voltage Booster Control Bit

Controls the power voltage booster.

1 (R/W): On

0 (R/W): Off (default)

When the power supply voltage (V_{DD}) is within the range from 1.8 V to 2.5 V, write 1 to PBON to turn the power voltage booster on. The power voltage booster doubles the V_{DD} voltage to generate V_{D2} for driving the LCD system voltage regulator. In addition, set VDSEL (D1) to 1 to drive the LCD system voltage regulator with V_{D2}. It is not necessary to generate V_{D2} when the power supply voltage (V_{DD}) is 2.5 V or more. In this case, the power voltage booster should be turned off to reduce current consumption.

Note: When the power voltage booster is turned on, the V_{D2} output voltage requires about 1 ms to stabilize. Do not switch the power source for the LCD system voltage regulator to V_{D2} until the stabilization time has elapsed.

4.6 Precautions

- Be sure to avoid using the V_{D1}, V_{D2}, and V_{C1}–V_{C5} pin outputs to drive external circuits.
- If V_{DD} is used as the power source for the LCD system voltage regulator when V_{DD} is 2.5 V or less, the V_{C1} to V_{C5} voltages cannot be generated within the specifications.
- When the operating mode is switched, the internal operating voltage requires 5 ms (max.) to stabilize. Flash memory programming should be started after the stabilization time has elapsed.
- When the power voltage booster is turned on, the V_{D2} output voltage requires about 1 ms to stabilize. Do not switch the power source for the LCD system voltage regulator to V_{D2} until the stabilization time has elapsed.
- Current consumption increases in heavy load protection mode, therefore do not set heavy load protection mode with software if unnecessary.

5 Initial Reset

5.1 Initial Reset Sources

The S1C17701 has three initial reset sources that initialize the internal circuits.

(1) #RESET pin (external initial reset)
(2) Key-entry reset using the P0 ports (P00–P03 pins) (software selectable external initial reset)
(3) Watchdog timer (software selectable internal initial reset)

Figure 5.1.1 shows the configuration of the initial reset circuit.

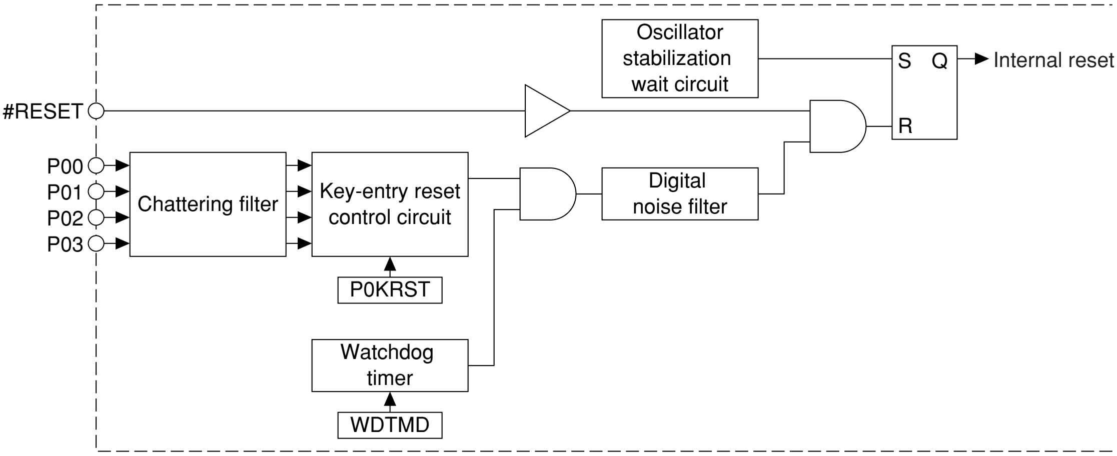


Figure 5.1.1 Configuration of Initial Reset Circuit

The CPU and peripheral circuits are initialized by the active signal from an initial reset source. When the reset signal is negated, the CPU starts reset handling. The reset handling reads the reset vector (reset handler start address) from the beginning of the vector table and starts executing the program (initial routine) beginning with the read address.

5.1.1 #RESET Pin

By setting the #RESET pin to low level, the S1C17701 enters initial reset state. In order to initialize the S1C17701 for sure, the #RESET pin must be held at low for more than the prescribed time (see Section 26.6, “AC Characteristics”) after the power supply voltage is supplied.
Initial reset state is canceled when the #RESET pin at low level is set to high level and the CPU starts executing the reset interrupt handler.
The #RESET pin is equipped with a pull-up resistor.

5.1.2 P0 Port Key-Entry Reset

Entering low level simultaneously to the ports (P00–P03) selected with software triggers an initial reset. The ports used for the reset function can be selected with the P0KRST[1:0] bits (D[1:0]/P0_KRST register).

∗ **P0KRST[1:0]**: P0 Port Key-Entry Reset Configuration Bits in the P0 Port Key-Entry Reset Configuration (P0_KRST) Register (D[1:0]/0x5209)

Table 5.1.2.1 Configuration of P0 Port Key-Entry Reset

P0KRST[1:0]	Port used for resetting
0x3	P00, P01, P02, P03
0x2	P00, P01, P02
0x1	P00, P01
0x0	Not used

For example, if P0KRST[1:0] is set to 0x3, an initial reset will take place when the four ports P00–P03 are set to low level at the same time.

Notes:
- When using the P0 port key-entry reset function, make sure that the designated input ports will not be simultaneously set to low level while the application program is running.
- The P0 port key-entry reset function cannot be used for power-on reset as it must be enabled with software.
- The P0 port key-entry reset function cannot be used in SLEEP mode.

5.1.3 Resetting by the Watchdog Timer

The S1C17701 has a built-in watchdog timer to detect runaway of the CPU. The watchdog timer overflows if it is not reset with software (due to CPU runaway) in four-second cycles. The overflow signal can generate either NMI or reset. Write 1 to the WDTMD bit (D1/WDT_ST register) to generate reset (NMI occurs when WDTMD = 0).

∗ **WDTMD**: NMI/Reset Mode Select Bit in the Watchdog Timer Status (WDT_ST) Register (D1/0x5041)

For details of the watchdog timer, see Chapter 17, “Watchdog Timer (WDT).”

Notes:
- When using the reset function of the watchdog timer, program the watchdog timer so that it will be reset within four-second cycles to avoid occurrence of an unnecessary reset.
- The reset function of the watchdog timer cannot be used for power-on reset as it must be enabled with software.

5.2 Initial Reset Sequence

Even if the #RESET pin input negates the reset signal after power is turned on, the CPU cannot boot up until the oscillation stabilization waiting time (1024/fOSC3 seconds*) has elapsed.
Figure 5.2.1 shows the operating sequence following cancellation of initial reset.
The CPU starts operating in synchronization with the OSC3 clock after reset state is canceled.

∗ fOSC3: OSC3 clock frequency

Note: The oscillation stabilization time described in this section does not include oscillation start time. Therefore the time interval until the CPU starts executing instructions after power is turned on or SLEEP mode is canceled may be longer than that indicated in the figure below.

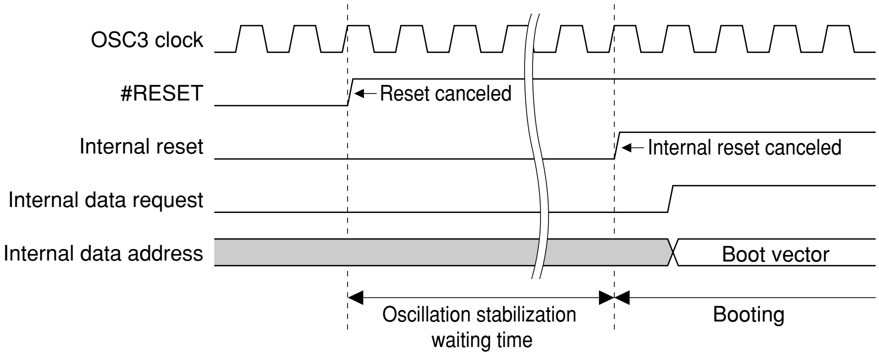


Figure 5.2.1 Operation Sequence Following Cancellation of Initial Reset

5.3 Initial Settings After an Initial Reset

The CPU internal registers are initialized as follows at initial reset.

R0–R7:	0x0
PSR:	0x0 (interrupt level = 0, interrupt disabled)
SP:	0x0
PC:	Reset vector stored at the beginning of the vector table is loaded by the reset handling.

The internal RAM and display memory should be initialized with software as they are not initialized at initial reset.

The internal peripheral modules are initialized to the default values (except some undefined registers). Change the settings with software if necessary. For the default values set at initial reset, see the list of I/O registers in Appendix or descriptions for each peripheral module.

6 Interrupt Controller (ITC)

6.1 Configuration of ITC

The S1C17701 provides 16 interrupt systems listed below.

1. P00–P07 input interrupt (8 types)
2. P10–P17 input interrupt (8 types)
3. Stopwatch timer interrupt (3 types)
4. Clock timer interrupt (4 types)
5. 8-bit OSC1 timer interrupt (1 type)
6. SVD interrupt (1 type)
7. LCD interrupt (1 type)
8. PWM & capture timer interrupt (2 types)
9. 8-bit timer interrupt (1 type)
10. 16-bit timer Ch.0 interrupt (1 type)
11. 16-bit timer Ch.1 interrupt (1 type)
12. 16-bit timer Ch.2 interrupt (1 type)
13. UART interrupt (3 types)
14. Remote controller interrupt (3 types)
15. SPI interrupt (2 types)
16. I^{2}C interrupt (2 types)

Each interrupt system provides an interrupt flag that indicates the occurrence of an interrupt request from the peripheral module and an interrupt enable bit that enables/disables interrupts. In addition, the ITC allows the application program to set the interrupt level (priority) of each interrupt system that determines the order of handling when two or more interrupts occur at the same time.
( ) in the list above represents the number of interrupt causes supported in each interrupt system. Use the control register in the peripheral module to select the interrupt causes for generating an interrupt request. For more information on interrupt causes and control, see the description for each peripheral module.
Figure 6.1.1 shows the structure of the interrupt system.

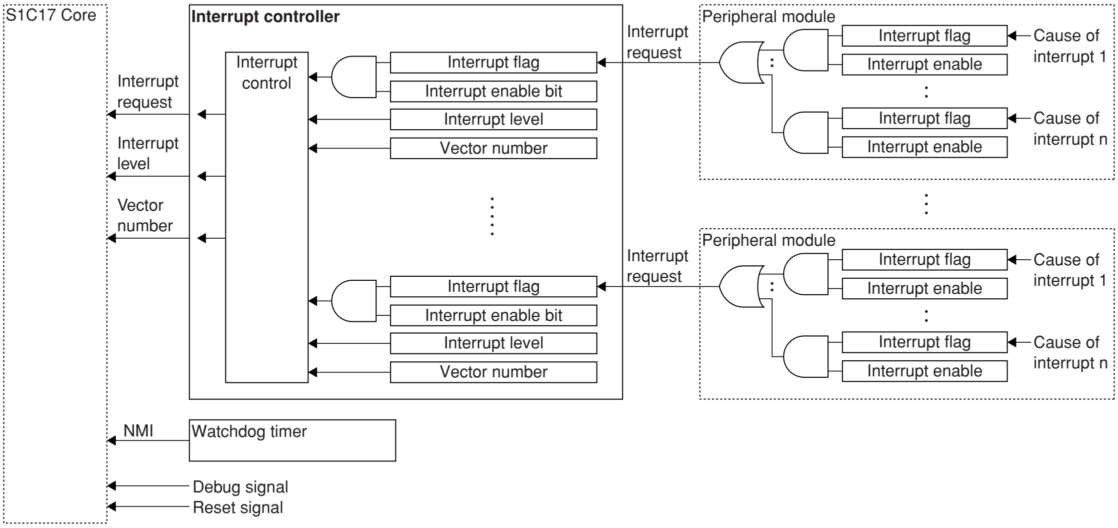


Figure 6.1.1 Interrupt System

6.2 Vector Table

The vector table contains the vectors to the interrupt handler routines (handler routine start address) that will be read by the S1C17 Core to execute the handler when an interrupt occurs.
The vector table is located at address 0x8000 in the S1C17701. The vector table base address can be read out from TTBR (Vector Table Base Register) located at address 0xffff80.
Table 6.2.1 shows the vector table of the S1C17701.

Table 6.2.1 Vector Table

Vector No. Software interrupt No.	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
0 (0x00)	0x8000	Reset	• Low input to the #RESET pin • Watchdog timer overflow *2	1
1 (0x01)	0x8004	Address misaligned interrupt	Memory access instruction	2
–	(0xfffc00)	Debugging interrupt	`brk` instruction, etc.	3
2 (0x02)	0x8008	NMI	Watchdog timer overflow *2	4
3 (0x03)	0x800c	reserved	–	–
4 (0x04)	0x8010	P0 port interrupt	P00–P07 port inputs	High *1 ↑
5 (0x05)	0x8014	P1 port interrupt	P10–P17 port inputs	
6 (0x06)	0x8018	Stopwatch timer interrupt	• 100 Hz timer signal • 10 Hz timer signal • 1 Hz timer signal	
7 (0x07)	0x801c	Clock timer interrupt	• 32 Hz timer signal • 8 Hz timer signal • 2 Hz timer signal • 1 Hz timer signal	
8 (0x08)	0x8020	8-bit OSC1 timer interrupt	Compare match	
9 (0x09)	0x8024	SVD interrupt	Low supply voltage detected	
10 (0x0a)	0x8028	LCD interrupt	Frame signal	
11 (0x0b)	0x802c	PWM & capture timer interrupt	• Compare match A • Compare match B	
12 (0x0c)	0x8030	8-bit timer interrupt	Timer underflow	
13 (0x0d)	0x8034	16-bit timer Ch.0 interrupt	Timer underflow	
14 (0x0e)	0x8038	16-bit timer Ch.1 interrupt	Timer underflow	
15 (0x0f)	0x803c	16-bit timer Ch.2 interrupt	Timer underflow	
16 (0x10)	0x8040	UART interrupt	• Transmit buffer empty • Receive buffer full • Receive error	
17 (0x11)	0x8044	Remote controller interrupt	• Data length counter underflow • Input rising edge detected • Input falling edge detected	
18 (0x12)	0x8048	SPI interrupt	• Transmit buffer empty • Receive buffer full	
19 (0x13)	0x804c	I^{2}C interrupt	• Transmit buffer empty • Receive buffer full	
20 (0x14) : 31 (0x1f)	0x8050 : 0x807c	reserved : reserved	– : –	↓ Low *1

*1 When the same interrupt level is set
*2 Either reset or NMI can be selected as the watchdog timer interrupt with software.

Vector numbers 4 to 19 are assigned to the maskable interrupts supported by the S1C17701.

6.3 Control of Maskable Interrupts

6.3.1 Enabling ITC

Before the ITC can be used, set the ITEN bit (D0/ITC_CTL register) to 1.

* **ITEN**: ITC Enable Bit in the ITC Control (ITC_CTL) Register (D0/0x4304)

6.3.2 Interrupt Request from Peripheral Module and Interrupt Flag

When an enabled interrupt cause occurs in a peripheral module, the module sends an interrupt request signal to the ITC. The interrupt request signal sets the interrupt flag in the ITC corresponding to the cause of interrupt to 1. The interrupt flag holds 1 until it is reset to 0 to indicate that an interrupt request has sent from the peripheral module. The flag status can be read from the ITC_IFLG register (0x4300).
Table 6.3.2.1 lists the relationship between the causes of interrupt and the interrupt flags.

Table 6.3.2.1 Causes of Hardware Interrupt and Interrupt Flags

Vector No.	Cause of hardware interrupt	Interrupt flag
4	P0 port interrupt: P00–P07 port inputs	EIFT0 (D0/ITC_IFLG register)
5	P1 port interrupt: P10–P17 port inputs	EIFT1 (D1/ITC_IFLG register)
6	Stopwatch timer interrupt: 100 Hz/10 Hz/1 Hz signal	EIFT2 (D2/ITC_IFLG register)
7	Clock timer interrupt: 32 Hz/8 Hz/2 Hz/1 Hz signal	EIFT3 (D3/ITC_IFLG register)
8	8-bit OSC1 timer interrupt: compare match	EIFT4 (D4/ITC_IFLG register)
9	SVD interrupt: low supply voltage detection	EIFT5 (D5/ITC_IFLG register)
10	LCD interrupt: frame signal	EIFT6 (D6/ITC_IFLG register)
11	PWM & capture timer interrupt: compare A/compare B match	EIFT7 (D7/ITC_IFLG register)
12	8-bit timer interrupt: timer underflow	IIFT0 (D8/ITC_IFLG register)
13	16-bit timer Ch.0 interrupt: timer underflow	IIFT1 (D9/ITC_IFLG register)
14	16-bit timer Ch.1 interrupt: timer underflow	IIFT2 (D10/ITC_IFLG register)
15	16-bit timer Ch.2 interrupt: timer underflow	IIFT3 (D11/ITC_IFLG register)
16	UART interrupt: transmit buffer empty/receive buffer full/receive error	IIFT4 (D12/ITC_IFLG register)
17	Remote controller interrupt: data length counter underflow/input rising edge/input falling edge	IIFT5 (D13/ITC_IFLG register)
18	SPI interrupt: transmit buffer empty/receive buffer full	IIFT6 (D14/ITC_IFLG register)
19	I^{2}C interrupt: transmit buffer empty/receive buffer full	IIFT7 (D15/ITC_IFLG register)

The ITC uses the interrupt flags to generate an interrupt to the S1C17 Core.
When an interrupt flag is set to 1, the ITC sends the interrupt request, interrupt level and vector number signals to the S1C17 Core if the interrupt has been enabled (see the next section).

The interrupt flag that has been set to 1 can be reset by writing 1. Reset the interrupt flag to 0 in the interrupt handler. If the interrupt handler does not reset the interrupt flag, the same interrupt will be generated again when the interrupt handling has finished (interrupts are disabled during interrupt handling and enabled by executing the `reti` instruction placed at the end of the interrupt handler).
Note, however, that the interrupt flags (EIFT0–EIFT7) for the level triggered interrupts (see Section 6.3.5) cannot be reset by writing 1. Those interrupt flags are reset when the interrupt signal is negated by the interrupt source.

For the occurrence conditions of the causes of interrupt and the module specific settings, refer to the section that describes the interrupt source module.

6.3.3 Enabling/Disabling Interrupts

To send an interrupt request to the S1C17 Core, the interrupt must be enabled one by one using the interrupt enable bit in the ITC_EN register (0x4302) corresponding to the interrupt flag. To enable an interrupt, set the interrupt enable bit to 1; to disable an interrupt, set the interrupt enable bit to 0 (default). The interrupt enable bit does not affect the interrupt flag status, so the interrupt flag will be set when an interrupt request from the peripheral module occurs regardless of how the interrupt enable bit is set.

Table 6.3.3.1 lists the correspondence between the interrupt enable bit and the interrupt flag.

Table 6.3.3.1 List of Interrupt Enable Bits

Vector No.	Hardware interrupt	Interrupt flag	Interrupt enable bit
4	P0 port interrupt	EIFT0 (D0/ITC_IFLG register)	EIEN0 (D0/ITC_EN register)
5	P1 port interrupt	EIFT1 (D1/ITC_IFLG register)	EIEN1 (D1/ITC_EN register)
6	Stopwatch timer interrupt	EIFT2 (D2/ITC_IFLG register)	EIEN2 (D2/ITC_EN register)
7	Clock timer interrupt	EIFT3 (D3/ITC_IFLG register)	EIEN3 (D3/ITC_EN register)
8	8-bit OSC1 timer interrupt	EIFT4 (D4/ITC_IFLG register)	EIEN4 (D4/ITC_EN register)
9	SVD interrupt	EIFT5 (D5/ITC_IFLG register)	EIEN5 (D5/ITC_EN register)
10	LCD interrupt	EIFT6 (D6/ITC_IFLG register)	EIEN6 (D6/ITC_EN register)
11	PWM & capture timer interrupt	EIFT7 (D7/ITC_IFLG register)	EIEN7 (D7/ITC_EN register)
12	8-bit timer interrupt	IIFT0 (D8/ITC_IFLG register)	IIEN0 (D8/ITC_EN register)
13	16-bit timer Ch.0 interrupt	IIFT1 (D9/ITC_IFLG register)	IIEN1 (D9/ITC_EN register)
14	16-bit timer Ch.1 interrupt	IIFT2 (D10/ITC_IFLG register)	IIEN2 (D10/ITC_EN register)
15	16-bit timer Ch.2 interrupt	IIFT3 (D11/ITC_IFLG register)	IIEN3 (D11/ITC_EN register)
16	UART interrupt	IIFT4 (D12/ITC_IFLG register)	IIEN4 (D12/ITC_EN register)
17	Remote controller interrupt	IIFT5 (D13/ITC_IFLG register)	IIEN5 (D13/ITC_EN register)
18	SPI interrupt	IIFT6 (D14/ITC_IFLG register)	IIEN6 (D14/ITC_EN register)
19	I^{2}C interrupt	IIFT7 (D15/ITC_IFLG register)	IIEN7 (D15/ITC_EN register)

Notes:

- To avoid unexpected interrupts being generated, always be sure to reset the interrupt flag before enabling the interrupt by writing 1 to the interrupt enable bit.
- In addition to the interrupt enable bit, the IE bit of the Processor Status Register (PSR) in the S1C17 Core must be set to 1 to actually generate an interrupt. If the IE bit has been set to 0, the S1C17 Core cannot accept a maskable interrupt request. In this case, the interrupt request sent from the ITC is held and it will be accepted after the IE bit is set to 1.

6.3.4 Processing when Multiple Interrupts Occur

The ITC provides the ITC_ELV*x* and ITC_ILV*x* registers (0x4306 to 0x4314) to set an interrupt level (zero to seven) for each cause of interrupt.

Table 6.3.4.1 Interrupt Level Setup Bits

Vector No.	Hardware interrupt	Interrupt level setup bits	Register address
4	P0 port interrupt	EILV0[2:0] (D[2:0]/ITC_ELV0 register)	0x4306
5	P1 port interrupt	EILV1[2:0] (D[10:8]/ITC_ELV0 register)	0x4306
6	Stopwatch timer interrupt	EILV2[2:0] (D[2:0]/ITC_ELV1 register)	0x4308
7	Clock timer interrupt	EILV3[2:0] (D[10:8]/ITC_ELV1 register)	0x4308
8	8-bit OSC1 timer interrupt	EILV4[2:0] (D[2:0]/ITC_ELV2 register)	0x430a
9	SVD interrupt	EILV5[2:0] (D[10:8]/ITC_ELV2 register)	0x430a
10	LCD interrupt	EILV6[2:0] (D[2:0]/ITC_ELV3 register)	0x430c
11	PWM & capture timer interrupt	EILV7[2:0] (D[10:8]/ITC_ELV3 register)	0x430c
12	8-bit timer interrupt	IILV0[2:0] (D[2:0]/ITC_ILV0 register)	0x430e
13	16-bit timer Ch.0 interrupt	IILV1[2:0] (D[10:8]/ITC_ILV0 register)	0x430e
14	16-bit timer Ch.1 interrupt	IILV2[2:0] (D[2:0]/ITC_ILV1 register)	0x4310
15	16-bit timer Ch.2 interrupt	IILV3[2:0] (D[10:8]/ITC_ILV1 register)	0x4310
16	UART interrupt	IILV4[2:0] (D[2:0]/ITC_ILV2 register)	0x4312
17	Remote controller interrupt	IILV5[2:0] (D[10:8]/ITC_ILV2 register)	0x4312
18	SPI interrupt	IILV6[2:0] (D[2:0]/ITC_ILV3 register)	0x4314
19	I^2C interrupt	IILV7[2:0] (D[10:8]/ITC_ILV3 register)	0x4314

The highest interrupt level is 7 and the lowest is 0.
The set interrupt level is sent to the S1C17 Core at the same time the ITC sends an interrupt request and is used by the S1C17 Core to disable subsequent interrupts that have the same or a lower interrupt level. (See Section 6.3.6 for more information.)
At initial reset, the interrupt levels are all set to 0. The S1C17 Core does not accept an interrupt request whose interrupt level is set to 0.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.
If two or more causes of interrupt that have been enabled by the interrupt enable bits occur simultaneously, the cause of interrupt whose ITC_ELV*x* or ITC_ILV*x* register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core.
If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first.
Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core.
If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to that of the new cause of interrupt. The first interrupt request is left pending.

6.3.5 Interrupt Trigger Mode

The ITC provides two trigger modes, the pulse trigger mode and the level trigger mode, to accept either a pulse signal or a level signal as interrupt requests from the interrupt sources that set EIFT flags.
The trigger mode can be selected using the EITG*x* bit in the ITC_ELV*x* registers (0x4306 to 0x430c). When EITG*x* is set to 1, level trigger mode is selected; when EITG*x* is set to 0 (default), pulse trigger mode is selected.

Note: Set all EITG*x* bits to 1 (level trigger mode) in the S1C17701.

Table 6.3.5.1 Trigger Mode Select Bits

Hardware interrupt	Trigger mode select bit	Register address
P0 port interrupt	EITG0 (D4/ITC_ELV0 register)	0x4306
P1 port interrupt	EITG1 (D12/ITC_ELV0 register)	0x4306
Stopwatch timer interrupt	EITG2 (D4/ITC_ELV1 register)	0x4308
Clock timer interrupt	EITG3 (D12/ITC_ELV1 register)	0x4308
8-bit OSC1 timer interrupt	EITG4 (D4/ITC_ELV2 register)	0x430a
SVD interrupt	EITG5 (D12/ITC_ELV2 register)	0x430a
LCD interrupt	EITG6 (D4/ITC_ELV3 register)	0x430c
PWM & capture timer interrupt	EITG7 (D12/ITC_ELV3 register)	0x430c

The interrupt source modules that set the IIFT flags output only a pulse signal to the ITC to request an interrupt, therefore, no trigger mode select bit is provided.

Pulse trigger mode

In pulse trigger mode, the ITC samples interrupt signals at the rising edge of the system clock. When a high pulse is sampled, the ITC sets the interrupt flag (IIFT*x*) to 1 and stops sampling of that interrupt signal. The ITC resumes the sampling operation for the interrupt signal after the interrupt flag (IIFT*x*) is reset to 0 in the application program (interrupt handler).

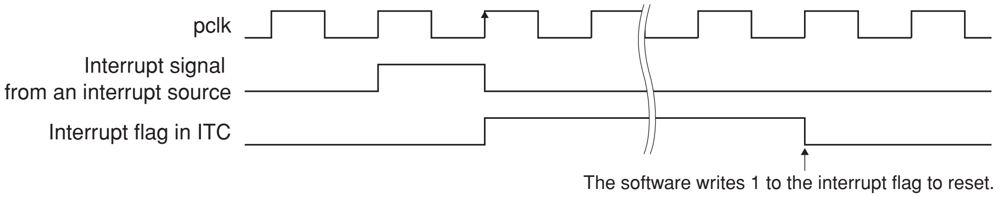

pclk
Interrupt signal from an interrupt source
Interrupt flag in ITC
The software writes 1 to the interrupt flag to reset.

Figure 6.3.5.1 Pulse Trigger Mode

Note: The following S1C17701 interrupts use pulse trigger mode. When an interrupt occurs, reset (write 1 to) the interrupt flag IIFT*x* in the interrupt handler routine.

- 8-bit timer interrupt
- 16-bit timer Ch.0 interrupt
- 16-bit timer Ch.1 interrupt
- 16-bit timer Ch.2 interrupt
- UART interrupt
- Remote controller interrupt
- SPI interrupt
- I^{2}C interrupt

Level trigger mode

In level trigger mode, the ITC continuously samples interrupt signals at every rising edge of the system clock. The interrupt flag (EIFT*x*) is set to 1 when a high level is sampled and is reset to 0 when a low level is sampled. In this mode, writing 1 cannot reset the interrupt flag (EIFT*x*). Therefore, the interrupt source module must hold the interrupt signal to high until the S1C17 Core accepts the interrupt request and must reset the interrupt signal after that.

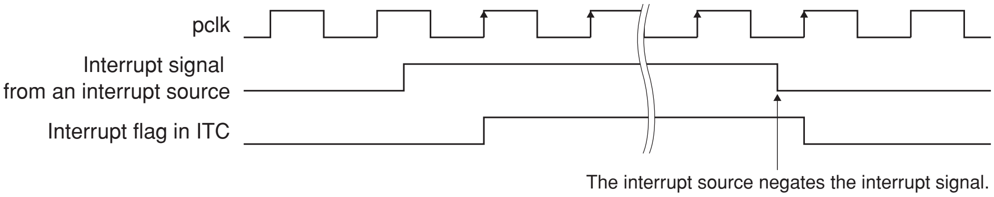


Figure 6.3.5.2 Level Trigger Mode

Note: The following S1C17701 interrupts use level trigger mode. The interrupt handler routine must reset (write 1 to) the interrupt flag provided in the peripheral module, not EIFT*x*.

- P0 port interrupt
- P1 port interrupt
- Stopwatch timer interrupt
- Clock timer interrupt
- 8-bit OSC1 timer interrupt
- SVD interrupt
- LCD interrupt
- PWM & capture timer interrupt

For the interrupt flag to be reset, see the description for each peripheral module.

6.3.6 Interrupt Processing by the S1C17 Core

A maskable interrupt to the S1C17 Core occurs when all of the conditions described below are met.

- The ITEN bit (D0/ITC_CTL register) is set to 1.

  * **ITEN**: ITC Enable Bit in the ITC Control (ITC_CTL) Register (D0/0x4304)

- The interrupt enable bit for the cause of interrupt that has occurred is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The cause of interrupt that has occurred has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

When a cause of interrupt occurs, the corresponding interrupt flag is set to 1 and the flag remains set until it is reset in the software program or by the hardware for a level triggered interrupt. Therefore, in no cases can the generated cause of interrupt be inadvertently cleared even if the above conditions are not met when the cause of interrupt has occurred. The interrupt will occur when the above conditions are met.
If two or more maskable causes of interrupt occur simultaneously, the cause of interrupt that has the highest priority is allowed to signal an interrupt request to the S1C17 Core. The other interrupts with lower priorities are kept pending until the above conditions are met.

The S1C17 Core keeps sampling interrupt requests every cycle. When the S1C17 Core accepts an interrupt request, it enters interrupt processing after completing execution of the instruction that was being executed.
The following lists the contents executed in interrupt processing.

(1) The PSR and the current program counter (PC) value are saved to the stack.
(2) The IE bit of the PSR is reset to 0 (following maskable interrupts are disabled).
(3) The IL of the PSR is set to the interrupt level of the accepted interrupt (NMI does not change the interrupt level).
(4) The vector of the interrupt occurred is loaded into the PC, thus executing the interrupt handler routine.

Thus, once an interrupt is accepted, all maskable interrupts that may follow are disabled in (2). Multiple interrupts can also be handled by setting the IE bit to 1 in the interrupt handler routine. In this case, since the IL has been changed in (3), only an interrupt that has a higher level than that of the currently processed interrupt is accepted.
When the interrupt handler routine is terminated by the `reti` instruction, the PSR is restored to its previous status before the interrupt has occurred. The program restarts processing after branching to the instruction next to the one that was being executed when the interrupt occurred.

6.4 NMI

In the S1C17701, the watchdog timer generates a non-maskable interrupt (NMI). The vector number of NMI is 2, with the vector address set to the vector table's starting address + 8 bytes.
This interrupt is prioritized over other interrupts and is unconditionally accepted by the S1C17 Core.

For how to generate NMI, see Chapter 17, “Watchdog Timer (WDT).”

6.5 Software Interrupts

The S1C17 Core provides the `int` *`imm5`* and `intl` *`imm5`*`,`*`imm3`* instructions allowing the software to generate any interrupts. The operand *imm5* specifies a vector number (0–31) in the vector table. In addition to this, the `intl` instruction has the operand *imm3* to specify the interrupt level (0–7) to be set to the IL field in the PSR.
The processor performs the same interrupt handling as that of the hardware interrupt.

6.6 Clearing HALT and SLEEP Modes by Interrupt Causes

A cause of interrupt clears HALT or SLEEP mode to start up the CPU.

The program execution sequence (whether it branches to the interrupt handler routine) after the CPU starts up depends on the clock status in HALT/SLEEP mode.

See “C.1 Power Saving by Clock Control” in Appendix C for details.

6.7 Details of Control Registers

Table 6.7.1 List of ITC Registers

Address	Register name		Function
0x4300	ITC_IFLG	Interrupt Flag Register	Indicates/resets interrupt occurrence status.
0x4302	ITC_EN	Interrupt Enable Register	Enables/disables each maskable interrupt.
0x4304	ITC_CTL	ITC Control Register	Enables/disables the ITC.
0x4306	ITC_ELV0	External Interrupt Level Setup Register 0	Sets the P0 and P1 interrupt levels and trigger modes.
0x4308	ITC_ELV1	External Interrupt Level Setup Register 1	Sets the stopwatch timer and clock timer interrupt levels and trigger modes.
0x430a	ITC_ELV2	External Interrupt Level Setup Register 2	Sets the 8-bit OSC1 timer and SVD interrupt levels and trigger modes.
0x430c	ITC_ELV3	External Interrupt Level Setup Register 3	Sets the LCD and PWM & capture timer interrupt levels and trigger modes.
0x430e	ITC_ILV0	Internal Interrupt Level Setup Register 0	Sets the 8-bit timer and 16-bit timer Ch.0 interrupt levels.
0x4310	ITC_ILV1	Internal Interrupt Level Setup Register 1	Sets the 16-bit timer Ch.1 and 16-bit timer Ch.2 interrupt levels.
0x4312	ITC_ILV2	Internal Interrupt Level Setup Register 2	Sets the UART and remote controller interrupt levels.
0x4314	ITC_ILV3	Internal Interrupt Level Setup Register 3	Sets the SPI and I^{2}C interrupt levels.

The following describes each ITC register. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all "reserved bits."

0x4300: Interrupt Flag Register (ITC_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Interrupt Flag Register (ITC_IFLG)	**0x4300** (16 bits)	D15	**IIFT7**	I^2C interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D14	**IIFT6**	SPI interrupt flag					0	R/W	
		D13	**IIFT5**	Remote controller interrupt flag					0	R/W	
		D12	**IIFT4**	UART interrupt flag					0	R/W	
		D11	**IIFT3**	16-bit timer Ch.2 interrupt flag					0	R/W	
		D10	**IIFT2**	16-bit timer Ch.1 interrupt flag					0	R/W	
		D9	**IIFT1**	16-bit timer Ch.0 interrupt flag					0	R/W	
		D8	**IIFT0**	8-bit timer interrupt flag					0	R/W	
		D7	**EIFT7**	PWM&capture timer interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1 in pulse trigger mode. Cannot be reset by software in level trigger mode.
		D6	**EIFT6**	LCD interrupt flag					0	R/W	
		D5	**EIFT5**	SVD interrupt flag					0	R/W	
		D4	**EIFT4**	8-bit OSC1 timer interrupt flag					0	R/W	
		D3	**EIFT3**	Clock timer interrupt flag					0	R/W	
		D2	**EIFT2**	Stopwatch timer interrupt flag					0	R/W	
		D1	**EIFT1**	P1 port interrupt flag					0	R/W	
		D0	**EIFT0**	P0 port interrupt flag					0	R/W	

D[15:8] IIFT[7:0]: Interrupt Flags (for Pulse Trigger)

These bits are interrupt flags to indicate the interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

The interrupt flag is set to 1 when a cause of interrupt occurs in each peripheral circuit.
If the following conditions are met at this time, an interrupt is generated to the S1C17 Core:

1. The corresponding bit of the Interrupt Enable Register is set to 1.
2. No other interrupt request of higher priority has occurred.
3. The IE bit of the PSR is set to 1 (interrupt enabled).
4. The corresponding interrupt level setup bits are set to a level higher than the S1C17 Core's interrupt level (IL).

The interrupt flag is always set to 1 when a cause of interrupt occurs regardless of how the interrupt enable and interrupt level setup bits are set.
In order for the next interrupt to be accepted after interrupt generation, the interrupt flag must be reset and the PSR must be set up again (by setting the IE bit to 1 or executing the `reti` instruction).
The flag that has been set to 1 can be reset by writing 1.

Table 6.7.2 Causes of Hardware Interrupt and Interrupt Flags

Interrupt flag	Cause of hardware interrupt
IIFT0 (D8)	8-bit timer interrupt: timer underflow
IIFT1 (D9)	16-bit timer Ch.0 interrupt: timer underflow
IIFT2 (D10)	16-bit timer Ch.1 interrupt: timer underflow
IIFT3 (D11)	16-bit timer Ch.2 interrupt: timer underflow
IIFT4 (D12)	UART interrupt: transmit buffer empty/receive buffer full/receive error
IIFT5 (D13)	Remote controller interrupt: data length counter underflow/input rising edge/ input falling edge
IIFT6 (D14)	SPI interrupt: transmit buffer empty/receive buffer full
IIFT7 (D15)	I^2C interrupt: transmit buffer empty/receive buffer full

D[7:0] EIFT[7:0]: Interrupt Flags (for Level Trigger)

These bits are interrupt flags to indicate the interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Has no effect
0 (W): Has no effect

See the description for IIFT[7:0].
However, these interrupts must be set to level trigger mode using the ITC_ELV*x* register (0x4306 to 0x430c). Therefore, EIFT*x* cannot be reset by writing 1. To reset the EIFT*x*, write 1 to the interrupt flag in the peripheral module.

Table 6.7.3 Causes of Hardware Interrupt and Interrupt Flags

Interrupt flag	Cause of hardware interrupt
EIFT0 (D0)	P0 port interrupt: P00–P07 port inputs
EIFT1 (D1)	P1 port interrupt: P10–P17 port inputs
EIFT2 (D2)	Stopwatch timer interrupt: 100 Hz/10 Hz/1 Hz signal
EIFT3 (D3)	Clock timer interrupt: 32 Hz/8 Hz/2 Hz/1 Hz signal
EIFT4 (D4)	8-bit OSC1 timer interrupt: compare match
EIFT5 (D5)	SVD interrupt: low supply voltage detection
EIFT6 (D6)	LCD interrupt: frame signal
EIFT7 (D7)	PWM & capture timer interrupt: compare A/compare B match

Note: Even when a maskable interrupt request is accepted by the S1C17 Core and control branches off to the interrupt handler routine, the interrupt flag is not reset. Consequently, if control is returned from the interrupt handler routine by the `reti` instruction without resetting the interrupt flag in a program, the same cause of interrupt occurs again. The interrupt flag of the level triggered interrupt must be reset using the control register in the peripheral module.

0x4302: Interrupt Enable Register (ITC_EN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Interrupt Enable Register (ITC_EN)	**0x4302** (16 bits)	D15	**IIEN7**	I^{2}C interrupt enable	1	Enable	0	Disable	0	R/W	
		D14	**IIEN6**	SPI interrupt enable					0	R/W	
		D13	**IIEN5**	Remote controller interrupt enable					0	R/W	
		D12	**IIEN4**	UART interrupt enable					0	R/W	
		D11	**IIEN3**	16-bit timer Ch.2 interrupt enable					0	R/W	
		D10	**IIEN2**	16-bit timer Ch.1 interrupt enable					0	R/W	
		D9	**IIEN1**	16-bit timer Ch.0 interrupt enable					0	R/W	
		D8	**IIEN0**	8-bit timer interrupt enable					0	R/W	
		D7	**EIEN7**	PWM&capture timer interrupt enable					0	R/W	
		D6	**EIEN6**	LCD interrupt enable					0	R/W	
		D5	**EIEN5**	SVD interrupt enable					0	R/W	
		D4	**EIEN4**	8-bit OSC1 timer interrupt enable					0	R/W	
		D3	**EIEN3**	Clock timer interrupt enable					0	R/W	
		D2	**EIEN2**	Stopwatch timer interrupt enable					0	R/W	
		D1	**EIEN1**	P1 port interrupt enable					0	R/W	
		D0	**EIEN0**	P0 port interrupt enable					0	R/W	

D[15:8] IIEN[7:0], EIEN[7:0]: Interrupt Enable Bits

These bits enable or disable interrupt generation.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Interrupts are enabled when the corresponding interrupt enable bit is set to 1 and are disabled when the bit is set to 0.

Table 6.7.4 Causes of Hardware Interrupt and Interrupt Enable Bits

Interrupt enable bits	Cause of hardware interrupt
EIEN0 (D0)	P0 port interrupt: P00–P07 port inputs
EIEN1 (D1)	P1 port interrupt: P10–P17 port inputs
EIEN2 (D2)	Stopwatch timer interrupt: 100 Hz/10 Hz/1 Hz signal
EIEN3 (D3)	Clock timer interrupt: 32 Hz/8 Hz/2 Hz/1 Hz signal
EIEN4 (D4)	8-bit OSC1 timer interrupt: compare match
EIEN5 (D5)	SVD interrupt: low supply voltage detection
EIEN6 (D6)	LCD interrupt: frame signal
EIEN7 (D7)	PWM & capture timer interrupt: compare A/compare B match
IIEN0 (D8)	8-bit timer interrupt: timer underflow
IIEN1 (D9)	16-bit timer Ch.0 interrupt: timer underflow
IIEN2 (D10)	16-bit timer Ch.1 interrupt: timer underflow
IIEN3 (D11)	16-bit timer Ch.2 interrupt: timer underflow
IIEN4 (D12)	UART interrupt: transmit buffer empty/receive buffer full/receive error
IIEN5 (D13)	Remote controller interrupt: data length counter underflow/input rising edge/ input falling edge
IIEN6 (D14)	SPI interrupt: transmit buffer empty/receive buffer full
IIEN7 (D15)	I^{2}C interrupt: transmit buffer empty/receive buffer full

0x4304: ITC Control Register (ITC_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
ITC Control Register (ITC_CTL)	**0x4304** (16 bits)	D15–1	**–**	reserved	–				–	–	0 when being read.
		D0	**ITEN**	ITC enable	1	Enable	0	Disable	0	R/W	

D[15:1] Reserved

D0 ITEN: ITC Enable Bit

Enables the ITC to control interrupt generation.

1 (R/W): Enable

0 (R/W): Disable (default)

Before the ITC can be used, this bit must be set to 1.

0x4306: External Interrupt Level Setup Register 0 (ITC_ELV0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
External Interrupt Level Setup Register 0 (ITC_ELV0)	**0x4306** (16 bits)	D15–13	**–**	reserved	–	–	–	0 when being read.
		D12	**EITG1**	P1 interrupt trigger mode	1 Level / 0 Pulse	0	R/W	Be sure to set to 1.
		D11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**EILV1[2:0]**	P1 interrupt level	0 to 7	0x0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**EITG0**	P0 interrupt trigger mode	1 Level / 0 Pulse	0	R/W	Be sure to set to 1.
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**EILV0[2:0]**	P0 interrupt level	0 to 7	0x0	R/W	

D[15:13] Reserved

D12 EITG1: P1 Port Interrupt Trigger Mode Select Bit

Selects the trigger mode of the P1 port interrupt. Set this bit 1 in the S1C17701.

1 (R/W): Level trigger mode

0 (R/W): Pulse trigger mode (default)

In pulse trigger mode, the ITC samples interrupt signals at the rising edge of the system clock. When a high pulse is sampled, the ITC sets the interrupt flag (EIFT*x*) to 1 and stops sampling of that interrupt signal. The ITC resumes the sampling operation for the interrupt signal after the interrupt flag (EIFT*x*) is reset to 0 in the application program (interrupt handler).

In level trigger mode, the ITC continuously samples interrupt signals at every rising edge of the system clock. The interrupt flag (EIFT*x*) is set to 1 when a high level is sampled and is reset to 0 when a low level is sampled. In this mode, writing 1 cannot reset the interrupt flag (EIFT*x*). Therefore, the interrupt source module must hold the interrupt signal to high until the S1C17 Core accepts the interrupt request and must reset the interrupt signal after that.

D11 Reserved

D[10:8] EILV1[2:0]: P1 Port Interrupt Level Bits

Sets the interrupt level (0 to 7) of the P1 port interrupt. (Default: 0)

If the level is set below the IL value of the PSR, the S1C17 Core does not accept the interrupt request.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.

If two or more causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt whose Interrupt Level Setup Register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core. If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first. Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core. If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to those of the new cause of interrupt. The first interrupt request is left pending.

D[7:5] Reserved

D4 EITG0: P0 Port Interrupt Trigger Mode Select Bit

Selects the trigger mode of the P0 port interrupt. Set this bit 1 in the S1C17701.

1 (R/W): Level trigger mode

0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12).

D3 Reserved

D[2:0] EILV0[2:0]: P0 Port Interrupt Level Bits

Sets the interrupt level (0 to 7) of the P0 port interrupt. (Default: 0)

See the description of EILV1[2:0] (D[10:8]).

0x4308: External Interrupt Level Setup Register 1 (ITC_ELV1)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
External Interrupt Level Setup Register 1 (ITC_ELV1)	**0x4308** (16 bits)	D15–13	–	reserved	–				–	–	0 when being read.
		D12	**EITG3**	CT interrupt trigger mode	1	Level	0	Pulse	0	R/W	Be sure to set to 1.
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV3[2:0]**	CT interrupt level	0 to 7				0x0	R/W	
		D7–5	–	reserved	–				–	–	0 when being read.
		D4	**EITG2**	SWT interrupt trigger mode	1	Level	0	Pulse	0	R/W	Be sure to set to 1.
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV2[2:0]**	SWT interrupt level	0 to 7				0x0	R/W	

D[15:13] Reserved

D12 EITG3: Clock Timer Interrupt Trigger Mode Select Bit

Selects the trigger mode of the clock timer interrupt. Set this bit 1 in the S1C17701.

1 (R/W): Level trigger mode

0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12) in the ITC_ELV0 register (0x4306).

D11 Reserved

D[10:8] EILV3[2:0]: Clock Timer Interrupt Level Bits

Sets the interrupt level (0 to 7) of the clock timer interrupt. (Default: 0)

See the description of EILV1[2:0] (D[10:8]) in the ITC_ELV0 register (0x4306).

D[7:5] Reserved

D4 EITG2: Stopwatch Timer Interrupt Trigger Mode Select Bit

Selects the trigger mode of the stopwatch timer interrupt. Set this bit 1 in the S1C17701.

1 (R/W): Level trigger mode

0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12) in the ITC_ELV0 register (0x4306).

D3 Reserved

D[2:0] EILV2[2:0]: Stopwatch Timer Interrupt Level Bits

Sets the interrupt level (0 to 7) of the stopwatch timer interrupt. (Default: 0)

See the description of EILV1[2:0] (D[10:8]) in the ITC_ELV0 register (0x4306).

0x430a: External Interrupt Level Setup Register 2 (ITC_ELV2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
External Interrupt Level Setup Register 2 (ITC_ELV2)	**0x430a** (16 bits)	D15–13	**–**	reserved	–	–	–	0 when being read.
		D12	**EITG5**	SVD interrupt trigger mode	1 Level 0 Pulse	0	R/W	Be sure to set to 1.
		D11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**EILV5[2:0]**	SVD interrupt level	0 to 7	0x0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**EITG4**	T8OSC1 interrupt trigger mode	1 Level 0 Pulse	0	R/W	Be sure to set to 1.
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**EILV4[2:0]**	T8OSC1 interrupt level	0 to 7	0x0	R/W	

D[15:13] Reserved

D12 EITG5: SVD Interrupt Trigger Mode Select Bit

Selects the trigger mode of the SVD interrupt. Set this bit 1 in the S1C17701.

1 (R/W): Level trigger mode

0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12) in the ITC_ELV0 register (0x4306).

D11 Reserved

D[10:8] EILV5[2:0]: SVD Interrupt Level Bits

Sets the interrupt level (0 to 7) of the SVD interrupt. (Default: 0)

See the description of EILV1[2:0] (D[10:8]) in the ITC_ELV0 register (0x4306).

D[7:5] Reserved

D4 EITG4: 8-bit OSC1 Timer Interrupt Trigger Mode Select Bit

Selects the trigger mode of the 8-bit OSC1 timer interrupt. Set this bit 1 in the S1C17701.

1 (R/W): Level trigger mode

0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12) in the ITC_ELV0 register (0x4306).

D3 Reserved

D[2:0] EILV4[2:0]: 8-bit OSC1 Timer Interrupt Level Bits

Sets the interrupt level (0 to 7) of the 8-bit OSC1 timer interrupt. (Default: 0)

See the description of EILV1[2:0] (D[10:8]) in the ITC_ELV0 register (0x4306).

0x430c: External Interrupt Level Setup Register 3 (ITC_ELV3)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
External Interrupt Level Setup Register 3 (ITC_ELV3)	**0x430c** (16 bits)	D15–13	–	reserved	–				–	–	0 when being read.
		D12	**EITG7**	T16E interrupt trigger mode	1	Level	0	Pulse	0	R/W	Be sure to set to 1.
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV7[2:0]**	T16E interrupt level	0 to 7				0x0	R/W	
		D7–5	–	reserved	–				–	–	0 when being read.
		D4	**EITG6**	LCD interrupt trigger mode	1	Level	0	Pulse	0	R/W	Be sure to set to 1.
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV6[2:0]**	LCD interrupt level	0 to 7				0x0	R/W	

D[15:13] Reserved

D12 EITG7: PWM & Capture Timer Interrupt Trigger Mode Select Bit

Selects the trigger mode of the PWM & capture timer interrupt. Set this bit 1 in the S1C17701.

1 (R/W): Level trigger mode

0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12) in the ITC_ELV0 register (0x4306).

D11 Reserved

D[10:8] EILV7[2:0]: PWM & Capture Timer Interrupt Level Bits

Sets the interrupt level (0 to 7) of the PWM & capture timer interrupt. (Default: 0)

See the description of EILV1[2:0] (D[10:8]) in the ITC_ELV0 register (0x4306).

D[7:5] Reserved

D4 EITG6: LCD Interrupt Trigger Mode Select Bit

Selects the trigger mode of the LCD interrupt. Set this bit 1 in the S1C17701.

1 (R/W): Level trigger mode

0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12) in the ITC_ELV0 register (0x4306).

D3 Reserved

D[2:0] EILV6[2:0]: LCD Interrupt Level Bits

Sets the interrupt level (0 to 7) of the LCD interrupt. (Default: 0)

See the description of EILV1[2:0] (D[10:8]) in the ITC_ELV0 register (0x4306).

0x430e: Internal Interrupt Level Setup Register 0 (ITC_ILV0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 0 (ITC_ILV0)	**0x430e** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV1[2:0]**	T16 Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV0[2:0]**	T8 interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] IILV1[2:0]: 16-bit Timer Ch.0 Interrupt Level Bits

Sets the interrupt level (0 to 7) of the 16-bit timer Ch.0 interrupt. (Default: 0)

If the level is set below the IL value of the PSR, the S1C17 Core does not accept the interrupt request.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.

If two or more causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt whose Interrupt Level Setup Register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core. If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first. Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core. If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to those of the new cause of interrupt. The first interrupt request is left pending.

D[7:3] Reserved

D[2:0] IILV0[2:0]: 8-bit Timer Interrupt Level Bits

Sets the interrupt level (0 to 7) of the 8-bit timer interrupt. (Default: 0)

See the description of IILV1[2:0] (D[10:8]).

0x4310: Internal Interrupt Level Setup Register 1 (ITC_ILV1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 1 (ITC_ILV1)	**0x4310** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV3[2:0]**	T16 Ch.2 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV2[2:0]**	T16 Ch.1 interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] IILV3[2:0]: 16-bit Timer Ch.2 Interrupt Level Bits

Sets the interrupt level (0 to 7) of the 16-bit timer Ch.2 interrupt. (Default: 0)
See the description of IILV1[2:0] (D[10:8]) in the ITC_ILV0 register (0x430e).

D[7:3] Reserved

D[2:0] IILV2[2:0]: 16-bit Timer Ch.1 Interrupt Level Bits

Sets the interrupt level (0 to 7) of the 16-bit timer Ch.1 interrupt. (Default: 0)
See the description of IILV1[2:0] (D[10:8]) in the ITC_ILV0 register (0x430e).

0x4312: Internal Interrupt Level Setup Register 2 (ITC_ILV2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 2 (ITC_ILV2)	**0x4312** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV5[2:0]**	REMC interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV4[2:0]**	UART interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] IILV5[2:0]: Remote Controller Interrupt Level Bits

Sets the interrupt level (0 to 7) of the remote controller interrupt. (Default: 0)

See the description of IILV1[2:0] (D[10:8]) in the ITC_ILV0 register (0x430e).

D[7:3] Reserved

D[2:0] IILV4[2:0]: UART Interrupt Level Bits

Sets the interrupt level (0 to 7) of the UART interrupt. (Default: 0)

See the description of IILV1[2:0] (D[10:8]) in the ITC_ILV0 register (0x430e).

0x4314: Internal Interrupt Level Setup Register 3 (ITC_ILV3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 3 (ITC_ILV3)	**0x4314** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV7[2:0]**	I^{2}C interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV6[2:0]**	SPI interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] IILV7[2:0]: I^{2}C Interrupt Level Bits

Sets the interrupt level (0 to 7) of the I^{2}C interrupt. (Default: 0)

See the description of IILV1[2:0] (D[10:8]) in the ITC_ILV0 register (0x430e).

D[7:3] Reserved

D[2:0] IILV6[2:0]: SPI Interrupt Level Bits

Sets the interrupt level (0 to 7) of the SPI interrupt. (Default: 0)

See the description of IILV1[2:0] (D[10:8]) in the ITC_ILV0 register (0x430e).

6.8 Precautions

- To prevent another interrupt from being generated for the same cause again after generation of an interrupt, be sure to reset the interrupt flag before enabling interrupts and setting the PSR again or executing the `reti` instruction.
- The following S1C17701 interrupts use level trigger mode.
  - P0 port interrupt
  - P1 port interrupt
  - Stopwatch timer interrupt
  - Clock timer interrupt
  - 8-bit OSC1 timer interrupt
  - SVD interrupt
  - LCD interrupt
  - PWM & capture timer interrupt

  Set all EITG*x* bits in the ITC_ELV*x* register (0x4306 to 0x430c) to 1 (level trigger mode).
  Furthermore, the interrupt handler routine must reset (write 1 to) the interrupt flag provided in the peripheral module, not EIFT*x*. For the interrupt flag to be reset, see the description for each peripheral module.

THIS PAGE IS BLANK.

7 Oscillator (OSC)

7.1 Configuration of OSC Module

The S1C17701 has two built-in oscillators (OSC3 and OSC1). The OSC3 oscillator generates the main clock (Max. 8.2 MHz) for operating the S1C17 Core and peripheral circuits at high speed. The OSC1 oscillator generates the sub clock (Typ. 32.768 kHz) for operating timers and for power saving operations.

At initial reset, the OSC3 clock is selected as the system clock.

The oscillators can be turned on and off and the system clock can be switched between OSC1 and OSC3 with software.

The OSC module allows the software to turn the oscillators on and off and to switch the system clock source between OSC1 and OSC3.

Furthermore, the clocks generated in the OSC module can be output outside the IC.

Figure 7.1.1 shows the structure of the clock system and the OSC3 module.

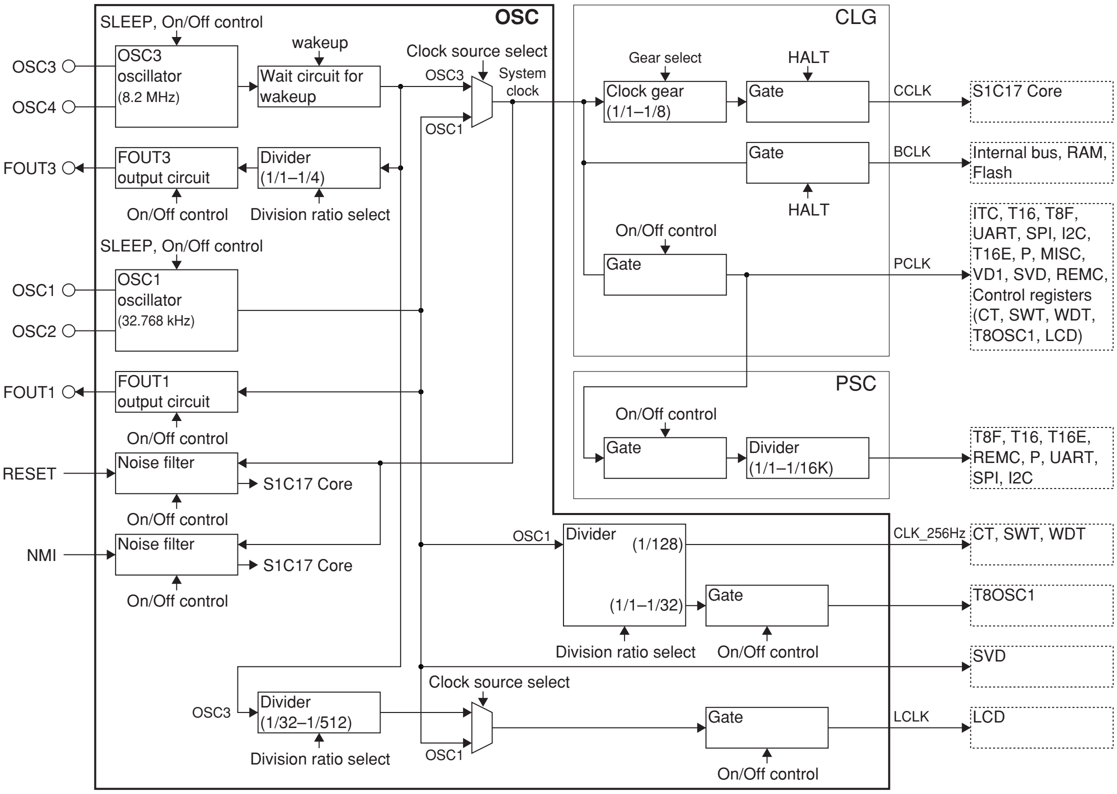


Figure 7.1.1 Structure of the OSC Module

Current consumption can be reduced by controlling the clocks according to the processing requirements as well as by using the standby mode. For methods to reduce current consumption, see Appendix C, "Power Saving."

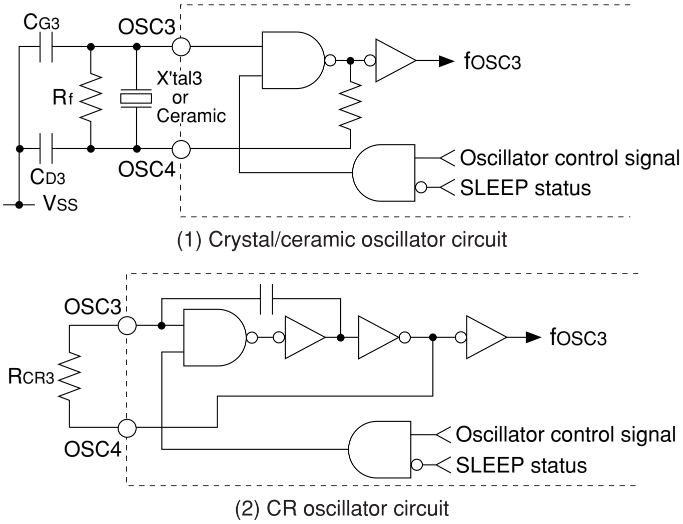
7.2 OSC3 Oscillator

The OSC3 oscillator generates the main clock (Max. 8.2 MHz) for operating the S1C17 Core and peripheral circuits at high speed. Depending on the product number, the oscillator type is crystal/ceramic oscillation (Max. 8.2 MHz) or CR oscillation (Max. 2.2 MHz).

Table 7.2.1 Lineup

Model No.	Main (OSC3) oscillator
S1C17701F00B100	Crystal/Ceramic
S1C17701F00E100	CR

Figure 7.2.1 shows the structure of the OSC3 oscillator circuit.



(1) Crystal/ceramic oscillator circuit

(2) CR oscillator circuit

Figure 7.2.1 OSC3 Oscillator Circuit

When the crystal/ceramic oscillator model is selected, connect a crystal (X'tal3) or ceramic resonator (Ceramic) and a feedback resistor (Rf) between the OSC3 and OSC4 pins, and two capacitors (C_{G3}, C_{D3}) to the OSC3 and OSC4 pins and Vss.
When the CR oscillator model is selected, connect only a resistor (R_{CR3}) between the OSC3 and OSC4 pins.

Controlling the OSC3 oscillation on and off

Setting OSC3EN (D0/OSC_CTL register) to 0 causes the OSC3 oscillator circuit to stop; setting it to 1 causes the OSC3 oscillator circuit to start oscillating. Also the OSC3 oscillator circuit stops when the S1C17 Core enters SLEEP mode.

∗ **OSC3EN**: OSC3 Enable Bit in the Oscillation Control (OSC_CTL) Register (D0/0x5061)

At initial reset, OSC3EN is set to 1 for enabling OSC3 oscillation. Furthermore, the OSC3 clock is selected as the system clock, so the S1C17 Core starts operating with the OSC3 clock.

Stable oscillation wait time when OSC3 starts oscillating

The OSC3 oscillator circuit provides an oscillation stabilization-wait timer to prevent malfunctions caused by an unstable clock immediately after the OSC3 oscillator starts oscillating such as when the power is turned on, when the S1C17 Core is woken from SLEEP mode or when software turns the OSC3 oscillator circuit on. The OSC3 clock supply is disabled until the time set to the timer has elapsed after the OSC3 oscillator starts oscillating. The stable oscillation wait time can be selected from four kinds of number of clock cycles using OSC3WT[1:0] (D[5:4]/OSC_CTL register).

∗ **OSC3WT[1:0]**: OSC3 Wait Cycle Select Bits in the Oscillation Control (OSC_CTL) Register (D[5:4]/0x5061)

Table 7.2.2 Setting Stable Oscillation Wait Time

OSC3WT[1:0]	Stable oscillation wait time
0x3	128 cycles
0x2	256 cycles
0x1	512 cycles
0x0	1024 cycles

(Default: 0x0)

The stable oscillation wait time is set to 1024 OSC3 clock cycles at initial reset, the S1C17 Core does not start operating until the set time has elapsed after releasing reset status.

Note: The oscillation start time varies depending on the resonator and externally attached parts. Set the stable oscillation wait time with a safety margin. Refer to the oscillation start time example described in Chapter 26, “Electrical Characteristics.”

7.3 OSC1 Oscillator

The OSC3 oscillator generates the 32.768 kHz (Typ.) sub-clock. Normally, the OSC1 clock is used as the operating clock for timers (clock timer, stopwatch timer, watchdog timer, and 8-bit OSC1 timer). Furthermore, it can be used as the system clock instead of the OSC3 clock to reduce current consumption when high-speed processing is not required.
The oscillator type is crystal oscillation.
Figure 7.3.1 shows the structure of the OSC1 oscillator circuit.

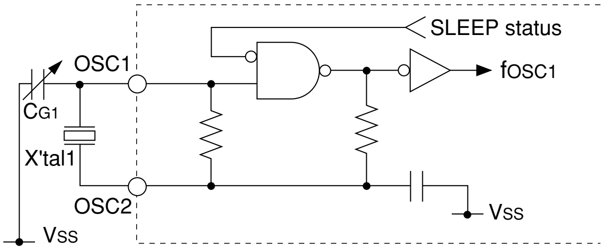


Figure 7.3.1 OSC1 Oscillator Circuit

To configure a crystal oscillator, connect a crystal X'tal1 (Typ. 32.768 kHz) between the OSC1 and OSC1 pins, and a trimmer capacitor C_{G1} (0–25 pF) between the OSC1 and VSS.

Controlling the OSC1 oscillation on and off

Setting OSC1EN (D1/OSC_CTL register) to 0 causes the OSC1 oscillator circuit to stop; setting it to 1 causes the OSC1 oscillator circuit to start oscillating. Also the OSC1 oscillator circuit stops when the S1C17 Core enters SLEEP mode.

* **OSC1EN**: OSC1 Enable Bit in the Oscillation Control (OSC_CTL) Register (D1/0x5061)

Stable oscillation wait time when OSC1 starts oscillating

The OSC1 oscillator circuit provides an oscillation stabilization-wait timer to prevent malfunctions caused by an unstable clock immediately after the OSC1 oscillator starts oscillating such as when the power is turned on, when the S1C17 Core is woken from SLEEP mode or when software turns the OSC1 oscillator circuit on. The OSC1 clock supply to the system is disabled for 256 OSC1 clock cycles after the OSC1 oscillator starts oscillating.

7.4 Switching the System Clock

The OSC module allows software to switch the system clock between the OSC3 and OSC1 clocks. Current consumption can be reduced by disabling the OSC3 oscillation after the system clock is switched to OSC1.
The following shows the control procedure:

OSC3 to OSC1

1. Set OSC1EN (D1/OSC_CTL register) to 1 to start the OSC1 oscillation if it is disabled.

   * **OSC1EN**: OSC1 Enable Bit in the Oscillation Control (OSC_CTL) Register (D1/0x5061)

2. Set CLKSRC (D0/OSC_SRC register) to 1 to switch the system clock from OSC3 to OSC1.

   * **CLKSRC**: System Clock Source Select Bit in the Clock Source Select (OSC_SRC) Register (D0/0x5060)

3. If the application does not need the peripheral modules clocked with OSC3 to operate, set OSC3EN (D0/OSC_CTL register) to 0 to stop the OSC3 oscillation.

   * **OSC3EN**: OSC3 Enable Bit in the Oscillation Control (OSC_CTL) Register (D0/0x5061)

Notes:
- When the system clock is switched from OSC3 to OSC1 immediately after the OSC1 oscillator starts oscillating, the system clock is halted until the OSC1 clock is activated (256 OSC1 clock-cycle period).
- The OSC3 oscillation cannot be stopped before switching the system clock to OSC1.

OSC1 to OSC3

1. Set a stable oscillation wait time (see Table 7.2.2) longer than the OSC3 oscillation start time using OSC3WT[1:0] (D[5:4]/OSC_CTL register). (This control is not necessary if it has been set already.)

   * **OSC3WT[1:0]**: OSC3 Wait Cycle Select Bits in the Oscillation Control (OSC_CTL) Register (D[5:4]/0x5061)

2. Set OSC3EN (D0/OSC_CTL register) to 1 to start the OSC3 oscillation if it is disabled. The OSC3 clock is not supplied to the system until the wait time set in OSC3WT[1:0] (D[5:4]/OSC_CTL register) has elapsed after the OSC3 oscillator starts oscillating.

3. Set CLKSRC (D0/OSC_SRC register) to 0 to switch the system clock from OSC1 to OSC3.

4. If the application does not need the peripheral modules clocked with OSC1 to operate, set OSC1EN (D1/OSC_CTL register) to 0 to stop the OSC1 oscillation.

Notes:
- Skip Steps 1 and 2 when the OSC2 oscillator circuit is operating.
- The OSC3 oscillation start time varies depending on the resonator and externally attached parts. Set the stable oscillation wait time with a safety margin. Refer to the oscillation start time example described in Chapter 26, “Electrical Characteristics.”
- The OSC1 oscillation cannot be stopped before switching the system clock to OSC3.

7.5 Controlling the LCD Clock

The OSC module incorporates the LCD clock generator to generate the operating clock (LCLK) for the LCD driver. See Chapter 22, “LCD Driver (LCD),” for details of the LCD driver.

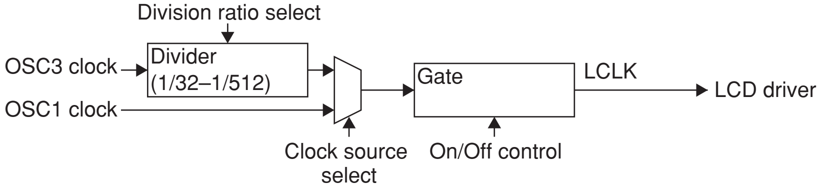


Figure 7.5.1 LCD Clock Generator

Selecting the source clock

Use LCKSRC (D1/OSC_LCLK register) to select either OSC1 or OSC3 as the source clock to generate the LCD clock. When LCKSRC is 1 (default), OSC1 is selected and when it is set to 1, OSC3 is selected.

∗ **LCKSRC**: LCD Clock Source Select Bit in the LCD Clock Setup (OSC_LCLK) Register (D1/0x5063)

Selecting a clock division ratio

When the OSC1 clock is used

When OSC1 is selected as the source clock, it is not necessary to select a division ratio. The OSC1 clock (Typ. 32.768 kHz) is sent directly to the LCD driver.

When the OSC3 clock is used

When OSC3 is selected as the source clock, select a division ratio using LCKDV[2:0] (D[4:2]/OSC_LCLK register).

∗ **LCKDV[2:0]**: LCD Clock Division Ratio Select Bits in the LCD Clock Setup (OSC_LCLK) Register (D[4:2]/0x5063)

Table 7.5.1 Selecting Division Ratio for LCD Clock

LCKDV[2:0]	Division ratio
0x7–0x5	Reserved
0x4	OSC3•1/512
0x3	OSC3•1/256
0x2	OSC3•1/128
0x1	OSC3•1/64
0x0	OSC3•1/32

(Default: 0x0)

Controlling the clock supply

Use LCKEN (D0/OSC_LCLK register) to control the clock supply to the LCD driver. LCKEN is set to 0 by default and the clock supply is disabled. When LCKEN is set to 1, the clock generated with the above conditions is supplied to the LCD driver. If display on the LCD is not necessary, disable the clock supply to reduce current consumption.

∗ **LCKEN**: LCD Clock Enable Bit in the LCD Clock Setup (OSC_LCLK) Register (D0/0x5063)

7.6 Controlling the 8-bit OSC1 Timer Clock

The OSC module incorporates a frequency divider and a clock supply control circuit for the 8-bit OSC1 timer. The 8-bit OSC1 timer is a programmable timer that operates with a divided OSC1 clock. See Chapter 14, "8-bit OSC1 Timer (T8OSC1)," for details of the 8-bit OSC1 timer.

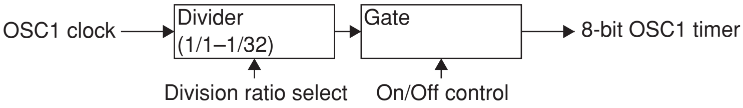


Figure 7.6.1 8-bit OSC1 Timer Clock Control Circuit

Selecting a clock division ratio

Select a clock division ratio of the OSC1 clock using T8O1CK[2:0] (D[3:1]/OSC_T8OSC1 register).

∗ **T8O1CK[2:0]**: T8OSC1 Clock Division Ratio Select Bits in the T8OSC1 Clock Control (OSC_T8OSC1) Register (D[3:1]/0x5065)

Table 7.6.1 Selecting Division Ratio for Generating T8OSC1 Clock

T8O1CK[2:0]	Division ratio
0x7–0x6	Reserved
0x5	OSC1•1/32
0x4	OSC1•1/16
0x3	OSC1•1/8
0x2	OSC1•1/4
0x1	OSC1•1/2
0x0	OSC1•1/1

(Default: 0x0)

Controlling the clock supply

Use T8O1CE (D0/OSC_T8OSC1 register) to control the clock supply to the 8-bit OSC1 timer. T8O1CE is set to 0 by default and the clock supply is disabled. When T8O1CE is set to 1, the clock generated with the above conditions is supplied to the 8-bit OSC1 timer. When the application does not need the 8-bit OSC1 timer to run, disable the clock supply to reduce current consumption.

∗ **T8O1CE**: T8OSC1 Clock Enable Bit in the T8OSC1 Clock Control (OSC_T8OSC1) Register (D0/0x5065)

7.7 External Output Clock (FOUT3, FOUT1)

A divided OSC3 clock (FOUT3) and the OSC1 clock (FOUT1) can be output to external devices.

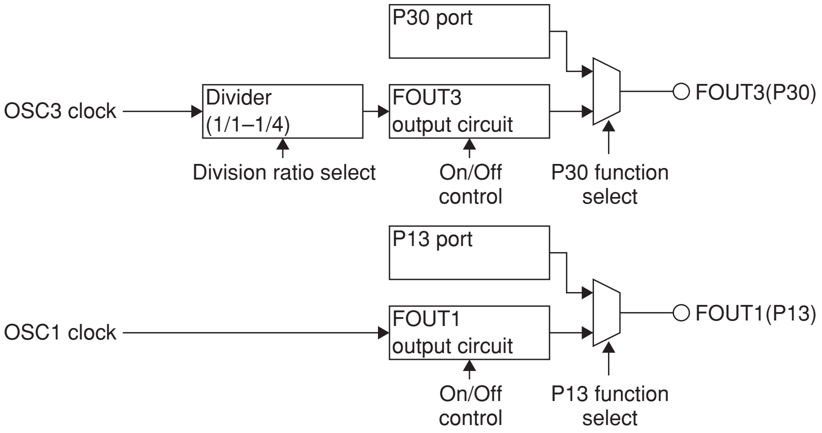


Figure 7.7.1 Clock Output Circuit

FOUT3 output

FOUT3 is a divided OSC3 clock.

Setting up the output pin

The FOUT3 output pin is shared with the P30 port and it functions as the P30 port pin by default. Write 1 to P30MUX (D0/P3_PMUX register) to switch the P30 pin function for the FOUT3 output.

∗ **P30MUX**: P30 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D0/0x52a3)

Selecting the FOUT3 clock frequency

The output clock frequency can be selected from three kinds. Select an OSC3 clock division ratio using FOUT3D[1:0] (D[3:2]/OSC_FOUT register) to set up the clock frequency.

∗ **FOUT3D[1:0]**: FOUT3 Clock Division Ratio Select Bits in the FOUT Control (OSC_FOUT) Register (D[3:2]/0x5064)

Table 7.7.1 Selecting Division Ratio for Generating FOUT3 Clock

FOUT3D[1:0]	Division ratio
0x3	Reserved
0x2	OSC3•1/4
0x1	OSC3•1/2
0x0	OSC3•1/1

(Default: 0x0)

Controlling the clock output

Use FOUT3E (D1/OSC_FOUT register) to control the clock output. When FOUT3E is set to 1, the FOUT3 clock is output from the FOUT3 pin and the output is disabled when FOUT3E is set to 0.

∗ **FOUT3E**: FOUT3 Output Enable Bit in the FOUT Control (OSC_FOUT) Register (D1/0x5064)

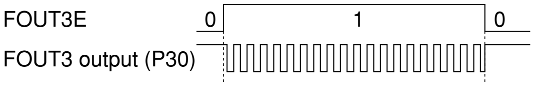


Figure 7.7.2 FOUT3 Output

Note: The FOUT3 signal is generated asynchronously with writing to FOUT3E, therefore, a hazard will occur when the output is enabled or disabled.

FOUT1 output

FOUT1 is the OSC1 clock.

Setting up the output pin

The FOUT1 output pin is shared with the P13 port and it functions as the P13 port pin by default. Write 1 to P13MUX (D3/P1_PMUX register) to switch the P13 pin function for the FOUT1 output.

* **P13MUX**: P13 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D3/0x52a1)

Controlling the clock output

Use FOUT1E (D0/OSC_FOUT register) to control the clock output. When FOUT1E is set to 1, the FOUT1 clock is output from the FOUT1 pin and the output is disabled when FOUT1E is set to 0.

* **FOUT1E**: FOUT1 Output Enable Bit in the FOUT Control (OSC_FOUT) Register (D0/0x5064)

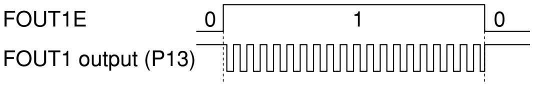


Figure 7.7.3 FOUT1 Output

Note: The FOUT1 signal is generated asynchronously with writing to FOUT1E, therefore, a hazard will occur when the output is enabled or disabled.

7.8 Noise Filters for RESET and NMI Inputs

If the RESET or NMI signal in the S1C17 Core input signals become active due to noise, the S1C17 Core executes unnecessary reset ot NMI handling. To avoid this, the OSC module incorporates noise filters that operate with the system clock to remove noise from these signals before they are input to the S1C17 Core.
The noise filter is provided for each signal, and can be enabled or bypassed individually.

RESET input noise filter: Noise will be removed when RSTFE (D1/OSC_NFEN register) = 1; the filter is bypassed when RSTFE = 0.
NMI input noise filter: Noise will be removed when NMIFE (D0/OSC_NFEN register) = 1; the filter is bypassed when NMIFE = 0.

∗ **RSTFE**: Reset Noise Filter Enable Bit in the Noise Filter Enable (OSC_NFEN) Register (D1/0x5062)
∗ **NMIFE**: NMI Noise Filter Enable Bit in the Noise Filter Enable (OSC_NFEN) Register (D0/0x5062)

The noise filter operates with a divide-by-8 system clock (OSC3 or OSC1 clock). When it is enabled, pulses that have a width of less than two cycles of this operating clock will be removed as noise. Therefore, 16 system clock cycles or longer pulse width is required to accept as a valid signal.

Notes:
- Enable the filter for the RESET input under normal circumstances.
- Although the S1C17701 has no external NMI input pin, the NMI request signal of the watchdog timer pass through the filter.

7.9 Details of Control Registers

Table 7.9.1 List of OSC Registers

Address	Register name		Function
0x5060	OSC_SRC	Clock Source Select Register	Selects a clock source.
0x5061	OSC_CTL	Oscillation Control Register	Controls oscillation.
0x5062	OSC_NFEN	Noise Filter Enable Register	Enables/disables noise filters.
0x5063	OSC_LCLK	LCD Clock Setup Register	Sets up the LCD clock.
0x5064	OSC_FOUT	FOUT Control Register	Controls clock output.
0x5065	OSC_T8OSC1	T8OSC1 Clock Control Register	Sets up the 8-bit OSC1 timer clock.

The following describes each OSC module control register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5060: Clock Source Select Register (OSC_SRC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Source Select Register (OSC_SRC)	**0x5060** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**CLKSRC**	System clock source select	1 OSC1 0 OSC3	0	R/W	

D[7:1] Reserved

D0 CLKSRC: System Clock Source Select Bit

Selects the system clock source.

1 (R/W): OSC1

0 (R/W): OSC3 (default)

Select OSC3 for normal (high-speed) operation. When the OSC3 clock is not necessary, select OSC1 as the system clock and stop OSC3 oscillation to reduce current consumption.

Note: When the system clock is switched from OSC3 to OSC1 immediately after the OSC1 oscillator starts oscillating, the system clock is halted until the OSC1 clock is activated (256 OSC1 clock-cycle period).

0x5061: Oscillation Control Register (OSC_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Oscillation Control Register (OSC_CTL)	**0x5061** (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–4	**OSC3WT[1:0]**	OSC3 wait cycle select	OSC3WT[1:0] Wait cycle 0x3 128 cycles 0x2 256 cycles 0x1 512 cycles 0x0 1024 cycles	0x0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	**OSC1EN**	OSC1 enable	1 Enable 0 Disable	1	R/W	
		D0	**OSC3EN**	OSC3 enable	1 Enable 0 Disable	1	R/W	

D[7:6] Reserved

D[5:4] OSC3WT[1:0]: OSC3 Wait Cycle Select Bits

Sets the stable oscillation wait time to avoid malfunctions caused by the unstable clock when the OSC3 starts oscillating.

The OSC3 clock is not supplied to the system until the wait time set here has elapsed after the OSC3 starts oscillating such as at power on, at wakeup from SLEEP status, or when the OSC3 oscillator is turned on with software.

Table 7.9.2 Setting the Stable OSC3 Oscillation Wait Time

OSC3WT[1:0]	Stable oscillation wait time
0x3	128 cycles
0x2	256 cycles
0x1	512 cycles
0x0	1024 cycles

(Default: 0x0)

At initial reset, the oscillation stabilization wait time is set to 1024 cycles (OSC3 clock). The CPU does not start operating until the set time has elapsed after the reset state is canceled.

Note: The oscillation start time will vary somewhat depending on the resonator and externally attached parts. Set the oscillation stabilization wait time allowing an adequate margin. For the oscillation start time, see an example indicated in Chapter 26, “Electrical Characteristics.”

D[3:2] Reserved

D1 OSC1EN: OSC1 Enable Bit

Enables/disables the OSC1 oscillator.

1 (R/W): Enable (On) (default)
0 (R/W): Disable (Off)

Notes:
- The OSC1 oscillator cannot be disabled when OSC1 is used as the system clock.
- In order to avoid malfunctions, the OSC1 clock will not be supplied to the system for 256 OSC1 clock-cycle period when the OSC1 oscillation is started by setting OSC1EN from 0 to 1.

D0 OSC3EN: OSC3 Enable Bit

Enables/disables the OSC3 oscillator.

1 (R/W): Enable (On) (default)
0 (R/W): Disable (Off)

Note: The OSC3 oscillator cannot be disabled when OSC3 is used as the system clock.

0x5062: Noise Filter Enable Register (OSC_NFEN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Noise Filter Enable Register (OSC_NFEN)	0x5062 (8 bits)	D7–2	–	reserved	–				–	–	0 when being read.
		D1	**RSTFE**	Reset noise filter enable	1	Enable	0	Disable	1	R/W	
		D0	**NMIFE**	NMI noise filter enable	1	Enable	0	Disable	0	R/W	

D[7:2] Reserved

D1 RSTFE: Reset Noise Filter Enable Bit

Enables/disables the noise filter for the RESET input.

1 (R/W): Enable (reject noise) (default)

0 (R/W): Disable (bypass)

When the noise filter is enabled, RESET pulses that have a width of 16 system clock (OSC1 or OSC3 clock) cycles or more will pass through the filter and are input to the S1C17 Core. Pulses that have a width of less than 16 cycles will be rejected as noise. Enable the filter under normal circumstances.

D0 NMIFE: NMI Noise Filter Enable Bit

Enables/disables the noise filter for the NMI input.

1 (R/W): Enable (reject noise)

0 (R/W): Disable (bypass) (default)

When the noise filter is enabled, NMI pulses that have a width of 16 system clock (OSC1 or OSC3 clock) cycles or more will pass through the filter and are input to the S1C17 Core. Pulses that have a width of less than 16 cycles will be rejected as noise.

Note: Although the S1C17701 has no external NMI input pin, the NMI request signal of the watchdog timer passes through the filter.

0x5063: LCD Clock Setup Register (OSC_LCLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
LCD Clock Setup Register (OSC_LCLK)	**0x5063** (8 bits)	D7–5	**–**	reserved	–		–	–	0 when being read.
		D4–2	**LCKDV[2:0]**	LCD clock division ratio select	LCKDV[2:0] 0x7–0x5 0x4 0x3 0x2 0x1 0x0	Division ratio reserved OSC3•1/512 OSC3•1/256 OSC3•1/128 OSC3•1/64 OSC3•1/32	0x0	R/W	
		D1	**LCKSRC**	LCD clock source select	1 OSC1	0 OSC3	1	R/W	
		D0	**LCKEN**	LCD clock enable	1 Enable	0 Disable	0	R/W	

D[7:5] Reserved

D[4:2] LCKDV[2:0]: LCD Clock Division Ratio Select Bits

Selects a division ratio when OSC3 is selected for the LCD clock source.

Table 7.9.3 Selecting the LCD Clock Division Ratio

LCKDV[2:0]	Division ratio
0x7–0x5	Reserved
0x4	OSC3•1/512
0x3	OSC3•1/256
0x2	OSC3•1/128
0x1	OSC3•1/64
0x0	OSC3•1/32

(Default: 0x0)

It is not necessary to select a division ratio when OSC1 is selected for the LCD clock source.

D1 LCKSRC: LCD Clock Source Select Bit

Select the LCD clock source.

1 (R/W): OSC1 (default)

0 (R/W): OSC3

D0 LCKEN: LCD Clock Enable Bit

Enables/disables supplying the LCD clock to the LCD driver.

1 (R/W): Enable (On)

0 (R/W): Disable (Off) (default)

LCKEN is set to 0 and the clock supply is disabled by default. By setting LCKEN to 1, the clock configured using the control bits above is supplied to the LCD driver. If an LCD display is unnecessary, disable the clock supply to reduce current consumption.

0x5064: FOUT Control Register (OSC_FOUT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
FOUT Control Register (OSC_FOUT)	**0x5064** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3–2	**FOUT3D[1:0]**	FOUT3 clock division ratio select	FOUT3D[1:0] 0x3 0x2 0x1 0x0		Division ratio reserved OSC3•1/4 OSC3•1/2 OSC3•1/1		0x0	R/W	
		D1	**FOUT3E**	FOUT3 output enable	1	Enable	0	Disable	0	R/W	
		D0	**FOUT1E**	FOUT1 output enable	1	Enable	0	Disable	0	R/W	

D[7:4] Reserved

D[3:2] FOUT3D[1:0]: FOUT3 Clock Division Ratio Select Bits

Selects a division ratio of the OSC3 clock to set the FOUT3 clock frequency.

Table 7.9.4 Selecting Division Ratio for FOUT3 Clock

FOUT3D[1:0]	Division ratio
0x3	Reserved
0x2	OSC3•1/4
0x1	OSC3•1/2
0x0	OSC3•1/1

(Default: 0x0)

D1 FOUT3E: FOUT3 Output Enable Bit

Enables/Disables the FOUT3 clock (OSC3 divide clock) to be output to a device outside the IC.

1 (R/W): Enable (On)

0 (R/W): Disable (Off) (default)

When FOUT3E is set to 1, the FOUT3 clock is output from the FOUT3 pin, and is stopped when FOUT3E is set to 0.

The FOUT3 pin is shared with the P30 port and it functions as the P30 port pin by default. When using the pin for the FOUT3 output, write 1 to the P30MUX bit (D0/P3_PMUX register) to switch the pin function.

∗ **P30MUX**: P30 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D0/0x52a3)

D0 FOUT1E: FOUT1 Output Enable Bit

Enables/Disables the FOUT1 clock (OSC1 clock) to be output to a device outside the IC.

1 (R/W): Enable (On)

0 (R/W): Disable (Off) (default)

When FOUT1E is set to 1, the FOUT1 clock is output from the FOUT1 pin, and is stopped when FOUT1E is set to 0.

The FOUT1 pin is shared with the P13 port and it functions as the P13 port pin by default. When using the pin for the FOUT1 output, write 1 to the P13MUX bit (D3/P1_PMUX register) to switch the pin function.

∗ **P13MUX**: P13 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D3/0x52a1)

0x5065: T8OSC1 Clock Control Register (OSC_T8OSC1)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
T8OSC1 Clock Control Register (OSC_T8OSC1)	0x5065 (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–1	T8O1CK[2:0]	T8OSC1 clock division ratio select	T8O1CK[2:0]	Division ratio	0x0	R/W	
					0x7–0x6	reserved			
					0x5	OSC1•1/32			
					0x4	OSC1•1/16			
					0x3	OSC1•1/8			
					0x2	OSC1•1/4			
					0x1	OSC1•1/2			
					0x0	OSC1•1/1			
		D0	T8O1CE	T8OSC1 clock output enable	1 Enable	0 Disable	0	R/W	

D[7:4] Reserved

D[3:1] T8O1CK[2:0]: T8OSC1 Clock Division Ratio Select Bits

Selects a division ratio of the OSC1 clock to configure the 8-bit OSC1 timer operating clock.

Table 7.9.5 Selecting Division Ratio for T8OSC1 Clock

T8O1CK[2:0]	Division ratio
0x7–0x6	Reserved
0x5	OSC1•1/32
0x4	OSC1•1/16
0x3	OSC1•1/8
0x2	OSC1•1/4
0x1	OSC1•1/2
0x0	OSC1•1/1

(Default: 0x0)

D0 T8O1CE: T8OSC1 Clock Output Enable Bit

Enables/disables supplying the operating clock to the 8-bit OSC1 timer.

1 (R/W): Enable (On)
0 (R/W): Disable (Off) (default)

T8O1CE is set to 0 and the clock supply is disabled by default. By setting T8O1CE to 1, the clock configured using the control bits above is supplied to the 8-bit OSC1 timer. If 8-bit OSC1 timer function is unnecessary, disable the clock supply to reduce current consumption.

7.10 Precautions

- The oscillation start time will vary somewhat depending on the resonator and externally attached parts. Set the OSC3 oscillation stabilization wait time allowing an adequate margin. For the oscillation start time, see an example indicated in Chapter 26, “Electrical Characteristics.”
- When the system clock is switched from OSC3 to OSC1 immediately after the OSC1 oscillator starts oscillating, the system clock is halted until the OSC1 clock is activated (256 OSC1 clock-cycle period).
- The OSC3 oscillator cannot be disabled when OSC3 is used as the system clock.
- The OSC1 oscillator cannot be disabled when OSC1 is used as the system clock.
- Since the FOUT3/FOUT1 signals are generated asynchronously with writing to FOUT3E/FOUT1E, a hazard may be generated when the signal is turned on or off.

8 Clock Generator (CLG)

8.1 Configuration of Clock Generator

The clock generator controls supplying the system clock to the S1C17 Core and peripheral modules.
Figure 8.1.1 shows the structure of the clock system and the CLG module.

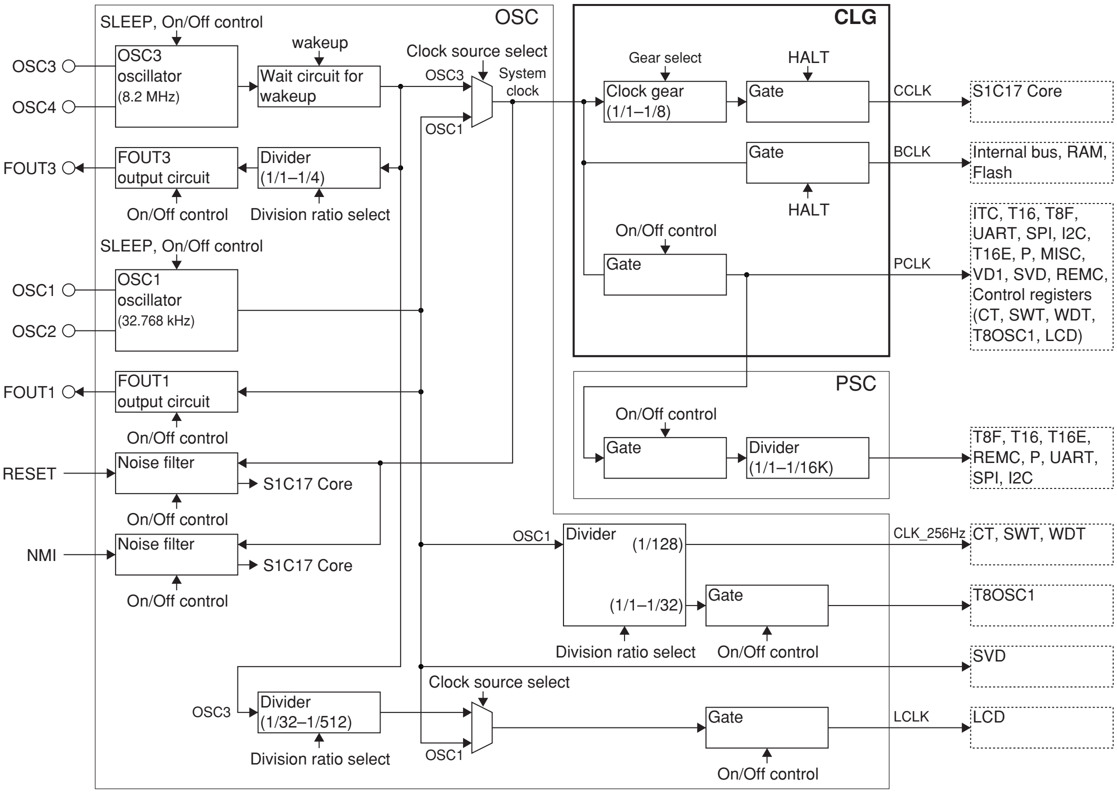


Figure 8.1.1 Structure of the CLG Module

Current consumption can be reduced by controlling the clocks according to the processing requirements as well as by using the standby mode. For methods to reduce current consumption, see Appendix C, “Power Saving.”

8.2 Controlling the CPU Core Clock (CCLK)

The CLG module incorporates clock gears to decelerate the system clock before supplying the clock to the S1C17 Core. Using a clock at as low a speed as possible to run the S1C17 Core reduces current consumption. Furthermore, the CLG stops clock supply to the S1C17 Core for power saving when the `halt` instruction is executed.

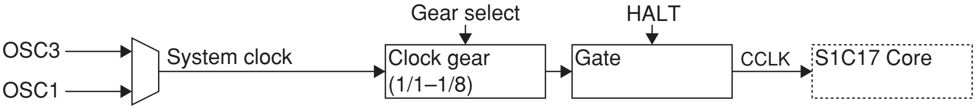


Figure 8.2.1 CCLK Supply System

Selecting a clock gear

Use CCLKGR[1:0] (D[1:0]/CLG_CCLK register) to select a gear for reducing the system clock speed.

∗ **CCLKGR[1:0]**: CCLK Clock Gear Ratio Select Bits in the CCLK Control (CLG_CCLK) Register (D[1:0]/0x5081)

Table 8.2.1 Selecting a CCLK Gear

CCLKGR[1:0]	Gear ratio
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

Controlling the clock supply

To stop the CCLK clock to be supplied to the S1C17 Core, execute the `halt` instruction. This does not stop the system clock, so the peripheral modules remain active.

HALT mode is canceled by occurrence of a reset, NMI, or other interrupt, and it resumes supplying CCLK.

Executing the `slp` instruction disables system clock supply to the CLG, therefore it also stops CCLK. When SLEEP mode is canceled by an external interrupt, supplying CCLK is resumed as well as the system clock supply to the CLG.

For control of the system clock, see Chapter 7, “Oscillator (OSC).”

8.3 Controlling the Peripheral Module Clock (PCLK)

The CLG module controls clock supply to the peripheral modules.
The system clock is used as the peripheral module clock (PCLK) without dividing.

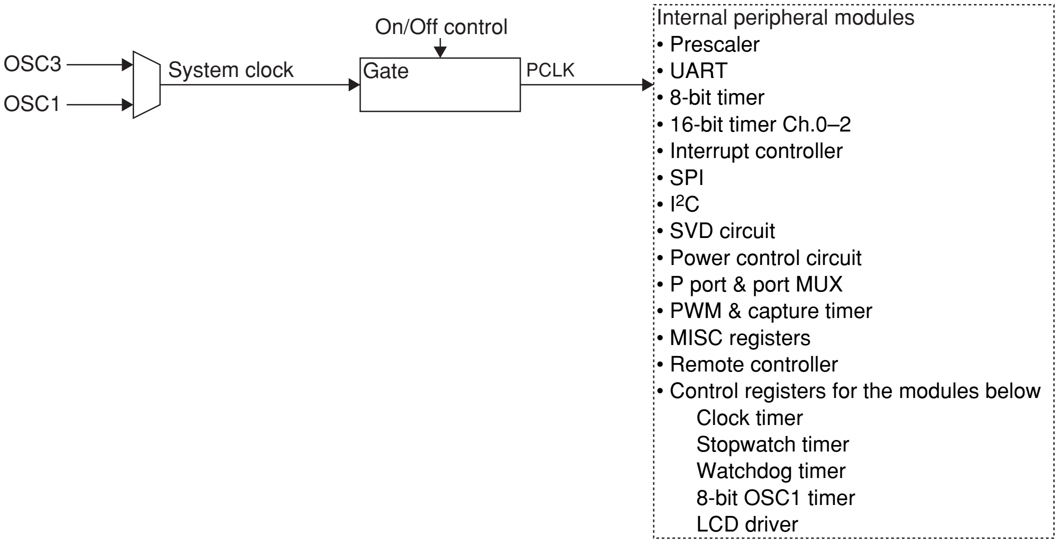


Figure 8.3.1 Peripheral Module Clock Control Circuit

Controlling the clock supply

Use PCKEN[1:0] (D[1:0]/CLG_PCLK register) to control supplying PCLK.

∗ **PCKEN[1:0]**: PCLK Enable Bits in the PCLK Control (CLG_PCLK) Register (D[1:0]/0x5080)

Table 8.3.1 PCLK Control

PCKEN[1:0]	PCLK supply
0x3	Enabled (On)
0x2	Not allowed
0x1	Not allowed
0x0	Disabled (Off)

(Default: 0x3)

These bits are set to 0x3 by default so that the clock is supplied to the peripheral modules. If all the peripheral modules in the internal peripheral area (listed above) can be idle, disable the clock supply to reduce current consumption.

Note: Be sure to avoid setting PCKEN[1:0] (D[1:0]/CLG_PCLK register) to 0x2 or 0x1, as some peripheral modules will stop operating.

Peripheral modules that operate with a clock other than PCLK

The peripheral modules shown below operate with a clock other than PCLK except for accessing the control registers. Therefore, PCLK is not necessary after the module starts operating by setting the control registers.

OSC1 peripheral modules

The clock timer, stopwatch timer, watchdog timer, and 8-bit OSC1 timer operate with a divided OSC1 clock. Although the control registers cannot be accessed for read and write when the PCLK supply is disabled, they keep operating.

LCD driver

The LCD driver operates with the OSC1 clock or a divided OSC3 clock. Although the control registers cannot be accessed for read and write when the PCLK supply is disabled, the LCD driver keeps refreshing the display. Also PCLK is not necessary for accessing the display memory.

8.4 Details of Control Registers

Table 8.4.1 List of CLG Registers

Address	Register name		Function
0x5080	CLG_PCLK	PCLK Control Register	Controls the PCLK output.
0x5081	CLG_CCLK	CCLK Control Register	Configures the CCLK division ratio.

The following describes each CLG module control register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5080: PCLK Control Register (CLG_PCLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
PCLK Control Register (CLG_PCLK)	**0x5080** (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	**PCKEN[1:0]**	PCLK enable	PCKEN[1:0]	PCLK supply	0x3	R/W	
					0x3	Enable			
					0x2	Not allowed			
					0x1	Not allowed			
					0x0	Disable			

D[7:2] Reserved

D[1:0] PCKEN[1:0]: PCLK Enable Bits

Enables/disables supplying the clock (PCLK) to the internal peripheral modules.

Table 8.4.2 PCLK Control

PCKEN[1:0]	PCLK supply
0x3	Enable (On)
0x2	Not allowed
0x1	Not allowed
0x0	Disable (Off)

(Default: 0x3)

PCKEN[1:0] is set to 0x3 and the clock supply is enabled by default. When the peripheral modules listed below are not used, disable the clock supply to reduce current consumption.

Peripheral modules that operate with PCLK
- Prescaler (PWM & capture timer, remote controller, P port)
- UART
- 8-bit timer
- 16-bit timer Ch.0–2
- Interrupt controller
- SPI
- I^{2}C
- SVD circuit
- Power control circuit
- P port & port MUX
- PWM & capture timer
- MISC register
- Remote controller

The peripheral modules listed below operate with a clock other than PCLK except for accessing their control registers. Therefore, PCLK is not required after the control registers are set once and the module starts operating.
- Clock timer
- Stopwatch timer
- Watchdog timer
- 8-bit OSC1 timer
- LCD driver

Note: Be sure to avoid setting PCKEN[1:0] to 0x2 or 0x1, as it stops some peripheral modules.

0x5081: CCLK Control Register (CLG_CCLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
CCLK Control Register (CLG_CCLK)	**0x5081** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**CCLKGR[1:0]**	CCLK clock gear ratio select	CCLKGR[1:0]	Gear ratio	0x0	R/W	
					0x3	1/8			
					0x2	1/4			
					0x1	1/2			
					0x0	1/1			

D[7:2] Reserved

D[1:0] CCLKGR[1:0]: CCLK Clock Gear Ratio Select Bits

Selects a gear ratio to decelerate the system clock. This determines the rate of the CCLK clock for driving the S1C17 Core. Drive the S1C17 Core with the slowest clock possible to reduce current consumption.

Table 8.4.3 Selecting CCLK Gear Ratio

CCLKGR[1:0]	Gear ratio
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

8.5 Precautions

- PCLK is supplied to the peripheral modules by default. If all the peripheral modules listed below can be placed in standby state, disable the clock supply to reduce current consumption.

  Peripheral modules that operate with PCLK
  - Prescaler (PWM & capture timer, remote controller, P port)
  - UART
  - 8-bit timer
  - 16-bit timer Ch.0–2
  - Interrupt controller
  - SPI
  - I^{2}C
  - SVD circuit
  - Power control circuit
  - P port & port MUX
  - PWM & capture timer
  - MISC register
  - Remote controller

  The peripheral modules listed below operate with a clock other than PCLK except for accessing their control registers. Therefore, PCLK is not required after the control registers are set once and the module starts operating.
  - Clock timer
  - Stopwatch timer
  - Watchdog timer
  - 8-bit OSC1 timer
  - LCD driver

- Be sure to avoid setting PCKEN[1:0] (D[1:0]/CLG_PCLK register) to 0x2 or 0x1, as it stops some peripheral modules.

  * **PCKEN[1:0]**: PCLK Enable Bits in the PCLK Control (CLG_PCLK) Register (D[1:0]/0x5080)

THIS PAGE IS BLANK.

9 Prescaler (PSC)

9.1 Configuration of the Prescaler

The S1C17701 incorporates a prescaler for generating the source clock of the timers and other peripheral modules. The prescaler divides the PCLK clock, which is supplied from the clock generator, by 1 to 16K to generate 15 clocks with different frequencies. A clock select register is provided for each destination peripheral module allowing selection of a prescaler output clock as the count or operating clock.

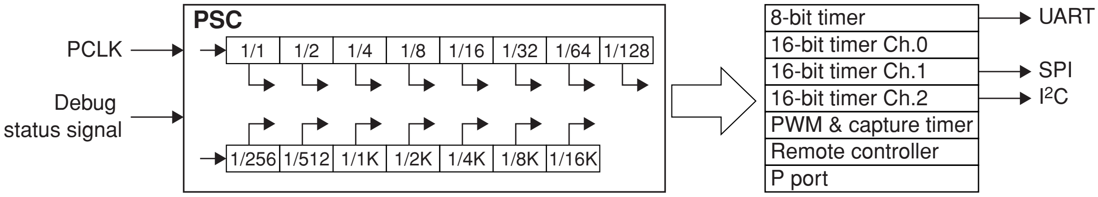


Figure 9.1.1 Prescaler

The prescaler is controlled by the PRUN bit (D0/PSC_CTL register). Write 1 to PRUN to run the prescaler and write 0 to stop the prescaler. When the timer and interface modules are idle, stop the prescaler to reduce current consumption. At initial reset, the prescaler stops operating.

∗ **PRUN**: Prescaler Run/Stop Control Bit in the Prescaler Control (PSC_CTL) Register (D0/0x4020)

Note: Supply PCLK from the clock generator before the prescaler can be used.

The prescaler provides one more control bit PRUND (D1/PSC_CTL register). This bit is used to specify the prescaler operation in debug mode. If PRUND is set to 1, the prescaler operates in debug mode. If PRUND is set to 0, the prescaler stops operating when the S1C17 Core enters debug mode. Set PRUND to 1 when using a timer or interface module in debug mode.

∗ **PRUND**: Prescaler Run/Stop Setting Bit in Debug Mode in the Prescaler Control (PSC_CTL) Register (D1/0x4020)

9.2 Details of Control Register

Table 9.2.1 Prescaler Register

Address	Register name		Function
0x4020	PSC_CTL	Prescaler Control Register	Starts/stops the prescaler.

The prescaler register is an 8-bit register.

Note: When setting the register, be sure to write a 0, and not a 1, for all "reserved bits."

0x4020: Prescaler Control Register (PSC_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Prescaler Control Register (PSC_CTL)	**0x4020** (8 bits)	D7–2	–	reserved	–				–	–	0 when being read.
		D1	**PRUND**	Prescaler run/stop in debug mode	1	Run	0	Stop	0	R/W	
		D0	**PRUN**	Prescaler run/stop control	1	Run	0	Stop	0	R/W	

D[7:2] Reserved

D1 PRUND: Prescaler Run/Stop Setting Bit for Debug Mode

Selects the prescaler operation in debug mode.

1 (R/W): Run

0 (R/W): Stop (default)

If PRUND is set to 1, the prescaler operates in debug mode. If PRUND is set to 0, the prescaler stops operating when the S1C17 Core enters debug mode. Set PRUND to 1 when using a timer or interface module in debug mode.

D0 PRUN: Prescaler Run/Stop Control Bit

Runs/stops the prescaler.

1 (R/W): Run

0 (R/W): Stop (default)

Write 1 to PRUN to run the prescaler and write 0 to stop the prescaler. When the timer and interface modules are idle, stop the prescaler to reduce current consumption.

9.3 Precaution

Supply PCLK from the clock generator before the prescaler can be used.

THIS PAGE IS BLANK.

10 I/O Ports (P)

10.1 Structure of I/O Port

The S1C17701 contains 28 I/O ports (P0[7:0], P1[7:0], P2[7:0], and P3[3:0]) that can be directed for input or output with software. Although the I/O pins, except for some I/O ports, are shared with internal peripheral modules, the pins can be used as general-purpose input/output ports unless they are used for the peripheral modules.
Figure 10.1.1 shows the structure of a typical I/O port.

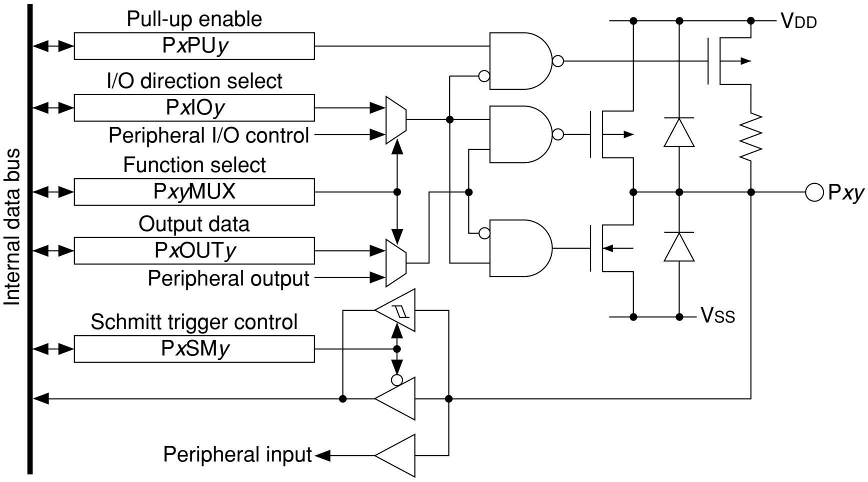


Figure 10.1.1 Structure of I/O Port

The P0 and P1 ports can generate input interrupts.
The P0[3:0] ports can be used as the key-entry reset ports (see Section 5.1.2, “P0 Port Key-Entry Reset,” for details).

Note: Supply PCLK from the clock generator before the I/O ports can be accessed.
Furthermore, a prescaler output clock is required for operating the P0 port chattering filter. Turn the prescaler on when using the chattering filter.

10.2 Selecting I/O Pin Functions (Port MUX)

The I/O port pins, except for some I/O ports, are shared with internal peripheral modules. Whether they are used as I/O ports or for peripheral modules can be selected using the port function select bit corresponding to each I/O port. All pins not used for peripheral modules can be used as general-purpose I/O ports.

Table 10.2.1 Selecting I/O Pin Functions

Pin function 1 P*xx*MUX = 0	Pin function 2 P*xx*MUX = 1	Port function select bit	Control register
P00	–	–	–
P01	–	–	–
P02	–	–	–
P03	–	–	–
P04	REMI (REMC)	P04MUX (D4)	P0 Port Function Select (P0_PMUX) Register (0x52a0)
P05	REMO (REMC)	P05MUX (D5)	
P06/**EXCL2** (T16CH2)	–	–	–
P07/**EXCL1** (T16CH1)	–	–	–
P10	–	–	–
P11	–	–	–
P12	–	–	–
P13	FOUT1 (OSC)	P13MUX (D3)	P1 Port Function Select (P1_PMUX) Register (0x52a1)
P14	SDA (I2C)	P14MUX (D4)	
P15	SCL (I2C)	P15MUX (D5)	
P16/**EXCL0** (T16CH0)	–	–	–
P17	#SPISS (SPI)	P17MUX (D7)	P1 Port Function Select (P1_PMUX) Register (0x52a1)
P20	SDI (SPI)	P20MUX (D0)	P2 Port Function Select (P2_PMUX) Register (0x52a2)
P21	SDO (SPI)	P21MUX (D1)	
P22	SPICLK (SPI)	P22MUX (D2)	
P23	SIN (UART)	P23MUX (D3)	
P24	SOUT (UART)	P24MUX (D4)	
P25	SCLK (UART)	P25MUX (D5)	
P26	TOUT (T16E)	P26MUX (D6)	
P27	EXCL3 (T16E)	P27MUX (D7)	
P30	FOUT3 (OSC)	P30MUX (D0)	P3 Port Function Select (P3_PMUX) Register (0x52a3)
DCLK (DBG)	P31	P31MUX (D1)	
DST2 (DBG)	P32	P32MUX (D2)	
DSIO (DBG)	P33	P33MUX (D3)	

At initial reset, each I/O port pin (P*xx*) is initialized for the default function ("Pin function 1" in Table 10.2.1).

The P06, P07, and P16 pins can be used as the external clock input pins for the 16-bit timer by setting the ports in input mode. However, general-purpose I/O function is also effective in this case, so no port function select bit is provided.

For the pin function other than the I/O port, see the descriptions for the peripheral module indicated in ( ).
The subsequent sections explain the port functions assuming that the pin has been set for the general-purpose I/O port.

10.3 Data Input/Output

The I/O ports allow selection of data input/output direction in bit units using the P*x*IO[7:0] bits (P*x*_IO register).

- ∗ **P0IO[7:0]**: P0[7:0] Port I/O Direction Select Bits in the P0 Port I/O Direction Control (P0_IO) Register (D[7:0]/0x5202)
- ∗ **P1IO[7:0]**: P1[7:0] Port I/O Direction Select Bits in the P1 Port I/O Direction Control (P1_IO) Register (D[7:0]/0x5212)
- ∗ **P2IO[7:0]**: P2[7:0] Port I/O Direction Select Bits in the P2 Port I/O Direction Control (P2_IO) Register (D[7:0]/0x5222)
- ∗ **P3IO[3:0]**: P3[3:0] Port I/O Direction Select Bits in the P3 Port I/O Direction Control (P3_IO) Register (D[3:0]/0x5232)

The input/output direction of the port configured for a peripheral module is controlled by the peripheral module and the bit setting in P*x*IO[7:0] is ignored.

Data input

To set an I/O port for input, set P*x*IO[7:0] to 0 (default). I/O ports set for input mode are placed in a high-impedance state, and thus function as input ports. The port pin is pulled up when the pull-up resistor is enabled using the P*x*_PU register.
In the input mode, the status of the input pin is read directly through P*x*IN[7:0] (P*x*_IN register), so the data is 1 when the pin status is a high (VDD) level or 0 when the pin status is a low (VSS) level.

- ∗ **P0IN[7:0]**: P0[7:0] Port Input Data Bits in the P0 Port Input Data (P0_IN) Register (D[7:0]/0x5200)
- ∗ **P1IN[7:0]**: P1[7:0] Port Input Data Bits in the P1 Port Input Data (P1_IN) Register (D[7:0]/0x5210)
- ∗ **P2IN[7:0]**: P2[7:0] Port Input Data Bits in the P2 Port Input Data (P2_IN) Register (D[7:0]/0x5220)
- ∗ **P3IN[3:0]**: P3[3:0] Port Input Data Bits in the P3 Port Input Data (P3_IN) Register (D[3:0]/0x5230)

Data output

To set an I/O port for output, set P*x*IO[7:0] to 1. I/O ports set for output function as output ports. When 1 is written to P*x*OUT[7:0] (P*x*_OUT register), the port outputs a high (VDD) level; when the data is 0, the port outputs a low (VSS) level. When the port is in output mode, the port pin is not pulled up even if the pull-up resistor is enabled with the P*x*_PU register.

- ∗ **P0OUT[7:0]**: P0[7:0] Port Output Data Bits in the P0 Port Output Data (P0_OUT) Register (D[7:0]/0x5201)
- ∗ **P1OUT[7:0]**: P1[7:0] Port Output Data Bits in the P1 Port Output Data (P1_OUT) Register (D[7:0]/0x5211)
- ∗ **P2OUT[7:0]**: P2[7:0] Port Output Data Bits in the P2 Port Output Data (P2_OUT) Register (D[7:0]/0x5221)
- ∗ **P3OUT[3:0]**: P3[3:0] Port Output Data Bits in the P3 Port Output Data (P3_OUT) Register (D[3:0]/0x5231)

Even in the input mode, data can be written to P*x*OUT[7:0] without affecting the pin status.

10.4 Pull-Up Control

The S1C17701 I/O ports have a built-in pull-up resistor and whether it is used or not can be selected using P*x*PU[7:0] (P*x*_PU register) for each bit individually.

- **P0PU[7:0]**: P0[7:0] Port Pull-up Enable Bits in the P0 Port Pull-up Control (P0_PU) Register (D[7:0]/0x5203)
- **P1PU[7:0]**: P1[7:0] Port Pull-up Enable Bits in the P1 Port Pull-up Control (P1_PU) Register (D[7:0]/0x5213)
- **P2PU[7:0]**: P2[7:0] Port Pull-up Enable Bits in the P2 Port Pull-up Control (P2_PU) Register (D[7:0]/0x5223)
- **P3PU[3:0]**: P3[3:0] Port Pull-up Enable Bits in the P3 Port Pull-up Control (P3_PU) Register (D[3:0]/0x5233)

The pull-up resistor is enabled by setting P*x*PU[7:0] to 1 (default), and the port pin will be pulled up when the port is in input mode. When set to 0, the pull-up resistor is disabled.
When the port is in output mode, the P*x*PU[7:0] settings are ignored and the pull-up resistor is disabled.

For unused ports, enable the pull-up resistors.
The pull-up settings are effective even if the port pin is configured for a peripheral module.

When changing the port pin from low level to high level with the built-in pull-up resistor, a delay in the waveform rise time will occur depending on the time constant of the pull-up resistor and the load capacitance of the pin. Therefore it is necessary to set an appropriate wait time for reading an I/O port. This wait time should be the amount of time or more calculated by the following expression.

Wait time = R_{IN} × (C_{IN} + load capacitance on the board) × 1.6 [seconds]
R_{IN}: Pull-up resistance Max. value
C_{IN}: Pin capacitance Max. value

10.5 Input Interface Level

The input interface level of the I/O port can be selected using P*x*SM[7:0] (P*x*_SM register) for each bit individually.

∗ **P0SM[7:0]**: P0[7:0] Port Schmitt Trigger Input Enable Bits in the P0 Port Schmitt Trigger Control (P0_SM) Register (D[7:0]/0x5204)
∗ **P1SM[7:0]**: P1[7:0] Port Schmitt Trigger Input Enable Bits in the P1 Port Schmitt Trigger Control (P1_SM) Register (D[7:0]/0x5214)
∗ **P2SM[7:0]**: P2[7:0] Port Schmitt Trigger Input Enable Bits in the P2 Port Schmitt Trigger Control (P2_SM) Register (D[7:0]/0x5224)
∗ **P3SM[3:0]**: P3[3:0] Port Schmitt Trigger Input Enable Bits in the P3 Port Schmitt Trigger Control (P3_SM) Register (D[3:0]/0x5234)

When P*x*SM[7:0] is set to 1 (default), the port is configured with a CMOS Schmitt level input interface. When the bit is set to 0, the port is configured with a CMOS level input interface.

10.6 Chattering Filter for P0 Ports

The P0[3:0] and P0[7:4] ports are equipped with a chattering filter and allow selection whether the filter is used or not. Furthermore, an input level check time can be selected when the filter is used. Use P0CF*x*[2:0] (P0_CHAT register) provided for the P0[3:0] port and P0[7:4] port individually to set the filtering conditions.

∗ **P0CF1[2:0]**: P0[3:0] Chattering Filter Time Select Bits in the P0 Port Chattering Filter Control (P0_CHAT) Register (D[2:0]/0x5208)

∗ **P0CF2[2:0]**: P0[7:4] Chattering Filter Time Select Bits in the P0 Port Chattering Filter Control (P0_CHAT) Register (D[6:4]/0x5208)

Table 10.6.1 Setting Input Level Check Time

P0CF*x*[2:0]	**Check time ∗**
0x7	16384/f_{PCLK} (8 ms)
0x6	8192/f_{PCLK} (4 ms)
0x5	4096/f_{PCLK} (2 ms)
0x4	2048/f_{PCLK} (1 ms)
0x3	1024/f_{PCLK} (512 µs)
0x2	512/f_{PCLK} (256 µs)
0x1	256/f_{PCLK} (128 µs)
0x0	Disabled (Off)

(Default: 0x0, ∗ when OSC3 = 2 MHz, PCLK = OSC3)

Notes:

- The check time to eliminate chattering means the maximum pulse width that can be eliminated. The valid interrupt input needs a pulse width of the set check time (minimum) to twice that of the check time (maximum).
- Input interrupts cannot be accepted in SLEEP mode if the CPU enters SLEEP mode when the chattering filter is active. The chattering filter should be disabled (off) before executing the `slp` instruction.
- Be sure to disable the P0 port interrupt before changing the P0_CHAT register (0x5208). Unnecessary interrupt may occur if the register is changed when the P0 port interrupt has been enabled.
- The internal signal may oscillate if the rise/fall time of the input signal is too long because the input signal level transition to the threshold level duration of time is too long. This causes the input interrupt to malfunction, therefore setup the input signal so that the rise/fall time is 25 ns or less.

10.7 Port Input Interrupt

The P0 and P1 ports provide an interrupt function.

Any ports within 16 ports can be selected for generating an interrupt. Furthermore, the interrupt condition, whether an interrupt will be generated at the rising edge or the falling edge, can be selected.

Figure 10.7.1 shows the configuration of the input interrupt circuit.

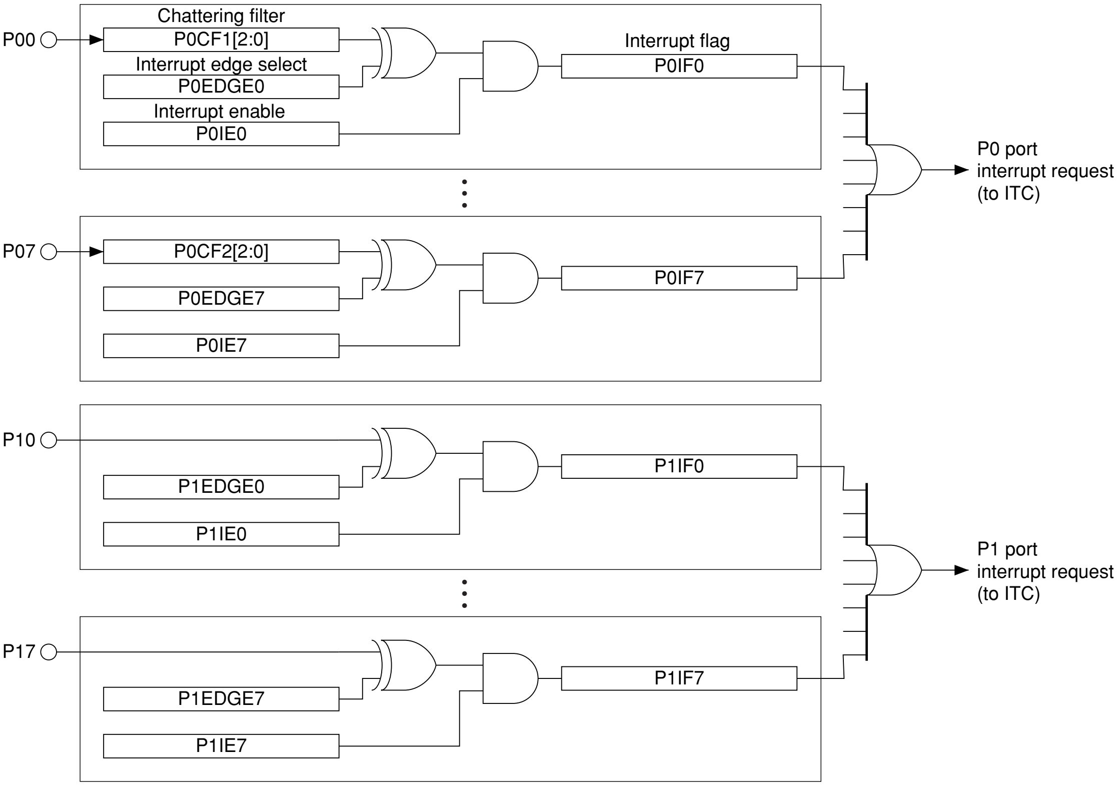


Figure 10.7.1 Configuration of Input Interrupt Circuit

Selecting interrupt ports

Select the ports to generate an interrupt using P*x*IE[7:0] (P*x*_IMSK register).

∗ **P0IE[7:0]**: P0[7:0] Port Interrupt Enable Bits in the P0 Port Interrupt Mask (P0_IMSK) Register (D[7:0]/0x5205)
∗ **P1IE[7:0]**: P1[7:0] Port Interrupt Enable Bits in the P1 Port Interrupt Mask (P1_IMSK) Register (D[7:0]/0x5215)

By setting P*x*IE[7:0] to 1, the corresponding port is enabled to generate an interrupt. The port whose P*x*IE bit is set to 0 (default) does not generate an interrupt.

In addition, it is necessary to set the interrupt controller to actually generate an interrupt. For setting the interrupt controller, refer to Chapter 6, "Interrupt Controller (ITC)."

Selecting interrupt edge

A port input interrupt can be generated either at the rising edge of the input signal or at the falling edge. Use P*x*EDGE[7:0] (P*x*_EDGE register) to select the input signal edge to generate an interrupt.

∗ **P0EDGE[7:0]**: P0[7:0] Port Interrupt Edge Select Bits in the P0 Port Interrupt Edge Select (P0_EDGE) Register (D[7:0]/0x5206)
∗ **P1EDGE[7:0]**: P1[7:0] Port Interrupt Edge Select Bits in the P1 Port Interrupt Edge Select (P1_EDGE) Register (D[7:0]/0x5216)

When P*x*EDGE[7:0] is set to 1, an input interrupt of the corresponding port will be generated at the falling edge; when the bit is set to 0 (default), an interrupt will be generated at the rising edge.

Interrupt flags in the P port module

Although the ITC accepts only two interrupt requests from the P0 and P1 ports, the P port module provides 16 interrupt flags P*x*IF[7:0] to control 16 port interrupts from P0[7:0] and P1[7:0] individually. P*x*IF[7:0] is set to 1 at a specified edge (rising edge or falling edge) of the input signal if the corresponding P*x*IE[7:0] has been set to 1. At the same time, a P0 or P1 interrupt request signal is output to the ITC. The interrupt request signal sets the P0 or P1 port interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

∗ **P0IF[7:0]**: P0[7:0] Port Interrupt Flags in the P0 Port Interrupt Flag (P0_IFLG) Register (D[7:0]/0x5207)
∗ **P1IF[7:0]**: P1[7:0] Port Interrupt Flags in the P1 Port Interrupt Flag (P1_IFLG) Register (D[7:0]/0x5217)

The settings shown below are required to manage the cause-of-interrupt occurrence status using the interrupt flags in the P port module.

1. Set the P0 and P1 interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the P*x*IF[7:0] interrupt flag of the P port module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

The P*x*IF[7:0] flags are reset by writing 1.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the P*x*IF[7:0] flags corresponding to the ports used to generate an interrupt before the interrupt is enabled using P*x*IE[7:0] (P*x*_IMSK register).

ITC registers for port interrupts

When the port whose interrupt is enabled detects the designated edge of the input signal according to the interrupt condition settings shown above, the P port module asserts the P0 or P1 port interrupt signal sent to the ITC.
To generate a port interrupt, set the interrupt level and enable the port interrupt using the ITC registers.
Table 10.7.1 lists the control bits for the port interrupt in the ITC.

Table 10.7.1 Control Bits in ITC

Port	Interrupt flag	Interrupt enable	Interrupt level setting	Trigger mode setting
P0	EIFT0 (D0/ITC_IFLG)	EIEN0 (D0/ITC_EN)	EILV0[2:0] (D[2:0]/ITC_ELV0)	EITG0 (D4/ITC_ELV0)
P1	EIFT1 (D1/ITC_IFLG)	EIEN1 (D1/ITC_EN)	EILV1[2:0] (D[10:8]/ITC_ELV0)	EITG1 (D12/ITC_ELV0)

ITC_IFLG register (0x4300)
ITC_EN register (0x4302)
ITC_ELV0 register (0x4306)

When the P0 or P1 port interrupt signal is asserted, the corresponding interrupt flag is set to 1. If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the timer interrupt, set the interrupt enable bit to 0. The interrupt flag is always set to 1 by the P0 or P1 port interrupt signal, regardless of how the interrupt enable bit is set (even when set to 0).
The interrupt level setup bits set the interrupt level (0 to 7) of the port interrupt. If the same interrupt level is set, the P0 port has higher priority.
As described above, the trigger mode setting bits for the port interrupts must be set to 1 (level trigger).

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The port interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to Chapter 6, “Interrupt Controller (ITC).”

Interrupt vectors

The following shows the vector numbers and vector addresses for the port interrupts:

Table 10.7.2 Port Interrupt Vectors

Port	Vector number	Vector address
P0	4 (0x04)	0x8010
P1	5 (0x05)	0x8014

10.8 Details of Control Registers

Table 10.8.1 List of I/O Port Registers

Address	Register name		Function
0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
0x5202	P0_IO	P0 Port I/O Direction Control Register	Selects the P0 port I/O direction.
0x5203	P0_PU	P0 Port Pull-up Control Register	Controls the P0 port pull-up resistor.
0x5204	P0_SM	P0 Port Schmitt Trigger Control Register	Controls the P0 port Schmitt trigger input.
0x5205	P0_IMSK	P0 Port Interrupt Mask Register	Enables/disables the P0 port interrupt.
0x5206	P0_EDGE	P0 Port Interrupt Edge Select Register	Selects the signal edge for generating P0 port interrupts.
0x5207	P0_IFLG	P0 Port Interrupt Flag Register	Indicates/resets the P0 port interrupt occurrence status.
0x5208	P0_CHAT	P0 Port Chattering Filter Control Register	Controls the P0 port chattering filter.
0x5209	P0_KRST	P0 Port Key-Entry Reset Configuration Register	Configures the P0 port key-entry reset function.
0x5210	P1_IN	P1 Port Input Data Register	P1 port input data
0x5211	P1_OUT	P1 Port Output Data Register	P1 port output data
0x5212	P1_IO	P1 Port I/O Direction Control Register	Selects the P1 port I/O direction.
0x5213	P1_PU	P1 Port Pull-up Control Register	Controls the P1 port pull-up resistor.
0x5214	P1_SM	P1 Port Schmitt Trigger Control Register	Controls the P1 port Schmitt trigger input.
0x5215	P1_IMSK	P1 Port Interrupt Mask Register	Enables/disables the P1 port interrupt.
0x5216	P1_EDGE	P1 Port Interrupt Edge Select Register	Selects the signal edge for generating P1 port interrupts.
0x5217	P1_IFLG	P1 Port Interrupt Flag Register	Indicates/resets the P1 port interrupt occurrence status.
0x5220	P2_IN	P2 Port Input Data Register	P2 port input data
0x5221	P2_OUT	P2 Port Output Data Register	P2 port output data
0x5222	P2_IO	P2 Port I/O Direction Control Register	Selects the P2 port I/O direction.
0x5223	P2_PU	P2 Port Pull-up Control Register	Controls the P2 port pull-up resistor.
0x5224	P2_SM	P2 Port Schmitt Trigger Control Register	Controls the P2 port Schmitt trigger input.
0x5230	P3_IN	P3 Port Input Data Register	P3 port input data
0x5231	P3_OUT	P3 Port Output Data Register	P3 port output data
0x5232	P3_IO	P3 Port I/O Direction Control Register	Selects the P3 port I/O direction.
0x5233	P3_PU	P3 Port Pull-up Control Register	Controls the P3 port pull-up resistor.
0x5234	P3_SM	P3 Port Schmitt Trigger Control Register	Controls the P3 port Schmitt trigger input.
0x52a0	P0_PMUX	P0 Port Function Select Register	Selects the P0 port function.
0x52a1	P1_PMUX	P1 Port Function Select Register	Selects the P1 port function.
0x52a2	P2_PMUX	P2 Port Function Select Register	Selects the P2 port function.
0x52a3	P3_PMUX	P3 Port Function Select Register	Selects the P3 port function.

The following describes each I/O port control register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5200/0x5210/0x5220/0x5230: P*x* Port Input Data Registers (P*x*_IN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Input Data Register (P0_IN)	**0x5200** (8 bits)	D7–0	**P0IN[7:0]**	P0[7:0] port input data	1	1 (H)	0	0 (L)	×	R	
P1 Port Input Data Register (P1_IN)	**0x5210** (8 bits)	D7–0	**P1IN[7:0]**	P1[7:0] port input data	1	1 (H)	0	0 (L)	×	R	
P2 Port Input Data Register (P2_IN)	**0x5220** (8 bits)	D7–0	**P2IN[7:0]**	P2[7:0] port input data	1	1 (H)	0	0 (L)	×	R	
P3 Port Input Data Register (P3_IN)	**0x5230** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3–0	**P3IN[3:0]**	P3[3:0] port input data	1	1 (H)	0	0 (L)	×	R	

Note: The letter '*x*' in bit names, etc., denotes a port number from 0 to 3.

D[7:0] P*x*IN[7:0]: P*x*[7:0] Port Input Data Bits (P3IN[3:0] for the P3 ports)

These bits are used to read data from I/O-port pins. (Default: external pin status)

1 (R): High level

0 (R): Low level

The PxIN[7:0] bits correspond to the Px[7:0] ports respectively and the voltage level on the port pin is read out in the input mode. If the pin voltage is high, 1 is read out as input data; if the pin voltage is low, 0 is read out as input data.
In the output mode, an indefinite value is read out.
P*x*IN[7:0] are read only bits and write operation is ineffective.

0x5201/0x5211/0x5221/0x5231: P*x* Port Output Data Registers (P*x*_OUT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Output Data Register (P0_OUT)	**0x5201** (8 bits)	D7–0	**P0OUT[7:0]**	P0[7:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P1 Port Output Data Register (P1_OUT)	**0x5211** (8 bits)	D7–0	**P1OUT[7:0]**	P1[7:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P2 Port Output Data Register (P2_OUT)	**0x5221** (8 bits)	D7–0	**P2OUT[7:0]**	P2[7:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P3 Port Output Data Register (P3_OUT)	**0x5231** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3–0	**P3OUT[3:0]**	P3[3:0] port output data	1	1 (H)	0	0 (L)	0	R/W	

Note: The letter '*x*' in bit names, etc., denotes a port number from 0 to 3.

D[7:0] **P*x*OUT[7:0]: P*x*[7:0] Port Output Data Bits** (P3OUT[3:0] for the P3 ports)

These bits are used to set data to be output from I/O-port pins.

1 (R/W): High level

0 (R/W): Low level (default)

The P*x*OUT[7:0] bits correspond to the P*x*[7:0] ports respectively and the data written to the register is directly output from the I/O port pin. If the data written to the port is 1, the port pin goes high; if the data is 0, the port pin goes low.

Even in input mode, data can be written to the port data register.

0x5202/0x5212/0x5222/0x5232: P*x* Port I/O Direction Control Registers (P*x*_IO)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port I/O Direction Control Register (P0_IO)	**0x5202** (8 bits)	D7–0	**P0IO[7:0]**	P0[7:0] port I/O direction select	1	Output	0	Input	0	R/W	
P1 Port I/O Direction Control Register (P1_IO)	**0x5212** (8 bits)	D7–0	**P1IO[7:0]**	P1[7:0] port I/O direction select	1	Output	0	Input	0	R/W	
P2 Port I/O Direction Control Register (P2_IO)	**0x5222** (8 bits)	D7–0	**P2IO[7:0]**	P2[7:0] port I/O direction select	1	Output	0	Input	0	R/W	
P3 Port I/O Direction Control Register (P3_IO)	**0x5232** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3–0	**P3IO[3:0]**	P3[3:0] port I/O direction select	1	Output	0	Input	0	R/W	

Note: The letter '*x*' in bit names, etc., denotes a port number from 0 to 3.

D[7:0] P*x*IO[7:0]: P*x*[7:0] Port I/O Direction Select Bits (P3IO[3:0] for the P3 ports)

Sets the I/O ports in input or output mode.

1 (R/W): Output mode

0 (R/W): Input mode (default)

The P*x*IO[7:0] bits are the input/output direction select bits corresponding to the P*x*[7:0] ports respectively. When a bit is set to 1, the corresponding I/O port is directed for output; if it is set to 0, the I/O port is directed for input.

When the pin is used for a peripheral module, the input/output direction depends on the peripheral function.

0x5203/0x5213/0x5223/0x5233: P*x* Port Pull-up Control Registers (P*x*_PU)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Pull-up Control Register (P0_PU)	**0x5203** (8 bits)	D7–0	**P0PU[7:0]**	P0[7:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	
P1 Port Pull-up Control Register (P1_PU)	**0x5213** (8 bits)	D7–0	**P1PU[7:0]**	P1[7:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	
P2 Port Pull-up Control Register (P2_PU)	**0x5223** (8 bits)	D7–0	**P2PU[7:0]**	P2[7:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	
P3 Port Pull-up Control Register (P3_PU)	**0x5233** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3–0	**P3PU[3:0]**	P3[3:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	

Note: The letter '*x*' in bit names, etc., denotes a port number from 0 to 3.

D[7:0] P*x*PU[7:0]: P*x*[7:0] Port Pull-up Enable Bits (P3PU[3:0] for the P3 ports)

Enables/disables the pull-up resistor incorporated in each port.

1 (R/W): Enable (default)

0 (R/W): Disable

The P*x*PU[7:0] bits are the pull-up control bits corresponding to the P*x*[7:0] ports respectively. When a bit is set to 1, the pull-up resistor is enabled and the corresponding I/O port is pulled up during input mode; if it is set to 0, the I/O port is not pulled up.
When the port is in output mode, the P*x*PU[7:0] settings are ignored and the port is not pulled up.
For unused ports, enable the pull-up resistors.
The pull-up settings are effective even if the port pin is configured for a peripheral module.

When changing the port pin from low level to high level with the built-in pull-up resistor, a delay in the waveform rise time will occur depending on the time constant of the pull-up resistor and the load capacitance of the pin. Therefore it is necessary to set an appropriate wait time for reading an I/O port. This wait time should be the amount of time or more calculated by the following expression.

Wait time = R_{IN} × (C_{IN} + load capacitance on the board) × 1.6 [seconds]

R_{IN}: Pull-up resistance Max. value

C_{IN}: Pin capacitance Max. value

0x5204/0x5214/0x5224/0x5234: P*x* Port Schmitt Trigger Control Registers (P*x*_SM)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Schmitt Trigger Control Register (P0_SM)	**0x5204** (8 bits)	D7–0	**P0SM[7:0]**	P0[7:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1 (0xff)	R/W	
P1 Port Schmitt Trigger Control Register (P1_SM)	**0x5214** (8 bits)	D7–0	**P1SM[7:0]**	P1[7:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1 (0xff)	R/W	
P2 Port Schmitt Trigger Control Register (P2_SM)	**0x5224** (8 bits)	D7–0	**P2SM[7:0]**	P2[7:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1 (0xff)	R/W	
P3 Port Schmitt Trigger Control Register (P3_SM)	**0x5234** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3–0	**P3SM[3:0]**	P3[3:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1 (0xff)	R/W	

Note: The letter ‘*x*’ in bit names, etc., denotes a port number from 0 to 3.

D[7:0] P*x*SM[7:0]: P*x*[7:0] Port Schmitt Trigger Input Enable Bits (P3SM[3:0] for the P3 ports)

Enables/disables the Schmitt trigger input buffer for each port.

1 (R/W): Enable (Schmitt input) (default)

0 (R/W): Disable (CMOS level)

The P*x*SM[7:0] bits are the Schmitt input control bits corresponding to the P*x*[7:0] ports respectively. When a bit is set to 1, the Schmitt trigger input buffer is enabled; if it is set to 0, the I/O port uses a CMOS level input buffer.

0x5205/5215: P*x* Port Interrupt Mask Registers (P*x*_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Interrupt Mask Register (P0_IMSK)	**0x5205** (8 bits)	D7–0	**P0IE[7:0]**	P0[7:0] port interrupt enable	1	Enable	0	Disable	0	R/W	
P1 Port Interrupt Mask Register (P1_IMSK)	**0x5215** (8 bits)	D7–0	**P1IE[7:0]**	P1[7:0] port interrupt enable	1	Enable	0	Disable	0	R/W	

Note: The letter '*x*' in bit names, etc., denotes a port number 0 or 1.

D[7:0] P*x*IE[7:0]: P*x*[7:0] Port Interrupt Enable Bits

Enables/disables the P0[7:0] and P1[7:0] port interrupts individually.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

Setting a P*x*IE[7:0] bit to 1 enables the interrupt of the corresponding port; setting to 0 disables the interrupt. The input level transition at the port pin whose interrupt has been disabled does not affect occurrence of the interrupt.

In addition, it is necessary to set the P0 and P1 port interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

0x5206/5216: P*x* Port Interrupt Edge Select Registers (P*x*_EDGE)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Interrupt Edge Select Register (P0_EDGE)	**0x5206** (8 bits)	D7–0	**P0EDGE[7:0]**	P0[7:0] port interrupt edge select	1	Falling edge	0	Rising edge	0	R/W	
P1 Port Interrupt Edge Select Register (P1_EDGE)	**0x5216** (8 bits)	D7–0	**P1EDGE[7:0]**	P1[7:0] port interrupt edge select	1	Falling edge	0	Rising edge	0	R/W	

Note: The letter ‘*x*’ in bit names, etc., denotes a port number 0 or 1.

D[7:0] P*x*EDGE[7:0]: P*x*[7:0] Port Interrupt Edge Select Bits

Selects an input signal edge to generate the P0[7:0] and P1[7:0] port interrupts individually.

1 (R/W): Falling edge
0 (R/W): Rising edge (default)

When a P*x*EDGE[7:0] bit is set to 1, an input interrupt of the corresponding port will be generated at the falling edge; when it is set to 0, an interrupt will be generated at the rising edge.

0x5207/5217: P*x* Port Interrupt Flag Registers (P*x*_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Interrupt Flag Register (P0_IFLG)	**0x5207** (8 bits)	D7–0	**P0IF[7:0]**	P0[7:0] port interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P1 Port Interrupt Flag Register (P1_IFLG)	**0x5217** (8 bits)	D7–0	**P1IF[7:0]**	P1[7:0] port interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

Note: The letter '*x*' in bit names, etc., denotes a port number 0 or 1.

D[7:0] P*x*IF[7:0]: P*x*[7:0] Port Interrupt Flags

These bits are interrupt flags to indicate the interrupt cause occurrence status.

1 (R):	Cause of interrupt has occurred
0 (R):	No cause of interrupt has occurred (default)
1 (W):	Flag is reset
0 (W):	Has no effect

The P*x*IF[7:0] bits are the interrupt flags corresponding to the 16 ports (P0[7:0], P1[7:0]) respectively. The interrupt flag is set to 1 at a specified edge (rising edge or falling edge) of the input signal if the corresponding P*x*IE[7:0] bit (P*x*_IMSK register) has been set to 1. At the same time, a P0 or P1 interrupt request signal is output to the ITC. The interrupt request signal sets the P0 or P1 port interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

The settings shown below are required to manage the cause-of-interrupt occurrence status using the interrupt flags in the P port module.

1. Set the P0 and P1 interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the P*x*IF[7:0] interrupt flag of the P port module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

The P*x*IF[7:0] flags are reset by writing 1.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the P*x*IF[7:0] flags corresponding to the ports used to generate an interrupt before the interrupt is enabled using P*x*IE[7:0] (P*x*_IMSK register).

∗ **P0IE[7:0]**: P0[7:0] Port Interrupt Enable Bits in the P0 Port Interrupt Mask (P0_IMSK) Register (D[7:0]/0x5205)

∗ **P1IE[7:0]**: P1[7:0] Port Interrupt Enable Bits in the P1 Port Interrupt Mask (P1_IMSK) Register (D[7:0]/0x5215)

0x5208: P0 Port Chattering Filter Control Register (P0_CHAT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P0 Port Chattering Filter Control Register (P0_CHAT)	**0x5208** (8 bits)	D7	–	reserved	–		–	–	0 when being read.
		D6–4	**P0CF2[2:0]**	P0[7:4] chattering filter time	P0CF2[2:0]	Filter time	0	R/W	
					0x7	16384/fPCLK	0x0	R/W	
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			
		D3	–	reserved	–		–	–	0 when being read.
		D2–0	**P0CF1[2:0]**	P0[3:0] chattering filter time	P0CF1[2:0]	Filter time	0x0	R/W	
					0x7	16384/fPCLK			
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			

D7 Reserved

D[6:4] P0CF2[2:0]: P0[7:4] Chattering Filter Time Select Bits

Configures the chattering filter for the P0[7:4] ports.

D3 Reserved

D[2:0] P0CF1[2:0]: P0[3:0] Chattering Filter Time Select Bits

Configures the chattering filter for the P0[3:0] ports.

The P0 ports are equipped with a chattering filter. Use P0CF*x*[2:0] to select whether the filter for P0[3:0] or P0[7:4] is used or not and also select an input level check time when the filter is used.

Table 10.8.2 Setting Input Level Check Time

P0CF*x*[2:0]	Check time *
0x7	16384/fPCLK (8 ms)
0x6	8192/fPCLK (4 ms)
0x5	4096/fPCLK (2 ms)
0x4	2048/fPCLK (1 ms)
0x3	1024/fPCLK (512 µs)
0x2	512/fPCLK (256 µs)
0x1	256/fPCLK (128 µs)
0x0	Disabled (Off)

(Default: 0x0, * when OSC3 = 2 MHz, PCLK = OSC3)

Notes:
- The check time to eliminate chattering means the maximum pulse width that can be eliminated. The valid interrupt input needs a pulse width of the set check time (minimum) to twice that of the check time (maximum).
- Input interrupts cannot be accepted in SLEEP mode if the CPU enters SLEEP mode when the chattering filter is active. The chattering filter should be disabled (off) before executing the `slp` instruction.
- Be sure to disable the P0 port interrupt before changing the P0_CHAT register. Unnecessary interrupt may occur if the register is changed when the P0 port interrupt has been enabled.
- The internal signal may oscillate if the rise/fall time of the input signal is too long because the input signal level transition to the threshold level duration of time is too long. This causes the input interrupt to malfunction, therefore setup the input signal so that the rise/fall time is 25 ns or less.

0x5209: P0 Port Key-Entry Reset Configuration Register (P0_KRST)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P0 Port Key-Entry Reset Configuration Register (P0_KRST)	**0x5209** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**P0KRST[1:0]**	P0 port key-entry reset configuration	P0KRST[1:0]	Configuration	0x0	R/W	
					0x3	P0[3:0] = 0			
					0x2	P0[2:0] = 0			
					0x1	P0[1:0] = 0			
					0x0	Disable			

D[7:2] Reserved

D[1:0] P0KRST[1:0]: P0 Port Key-Entry Reset Configuration Bits

Selects a port configuration for the P0 port key-entry reset function.

Table 10.8.3 Configuration of P0 Port Key-Entry Reset

P0KRST[1:0]	Port used for resetting
0x3	P00, P01, P02, P03
0x2	P00, P01, P02
0x1	P00, P01
0x0	Not used

(Default: 0x0)

The P0 key-entry reset is one of the initial reset features and it generates an initial reset signal when the ports selected here is set to low level at the same time.

For example, if P0KRST[1:0] is set to 0x3, an initial reset will take place when the four ports P00–P03 are set to low level at the same time.

Set P0KRST[1:0] to 0x0 when this function is not used.

Notes:
- When using the P0 port key-entry reset function, make sure that the designated input ports will not be simultaneously set to low level while the application program is running.
- The P0 port key-entry reset function cannot be used for power-on reset as it must be enabled with software.
- The P0 port key-entry reset function cannot be used in SLEEP mode.

0x52a0: P0 Port Function Select Register (P0_PMUX)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Function Select Register (P0_PMUX)	**0x52a0** (8 bits)	D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**P05MUX**	P05 port function select	1	REMO	0	P05	0	R/W	
		D4	**P04MUX**	P04 port function select	1	REMI	0	P04	0	R/W	
		D3–0	**–**	reserved	–				–	–	0 when being read.

The P04 and P05 I/O port pins are shared with a peripheral module. This register configures the pin functions.

D[7:6] Reserved

D5 P05MUX: P05 Port Function Select Bit

1 (R/W): REMO (REMC)

0 (R/W): P05 port (default)

D4 P04MUX: P04 Port Function Select Bit

1 (R/W): REMI (REMC)

0 (R/W): P04 port (default)

D[3:0] Reserved

0x52a1: P1 Port Function Select Register (P1_PMUX)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P1 Port Function Select Register (P1_PMUX)	**0x52a1** (8 bits)	D7	**P17MUX**	P17 port function select	1	#SPISS	0	P17	0	R/W	
		D6	**–**	reserved	–				–	–	0 when being read.
		D5	**P15MUX**	P15 port function select	1	SCL	0	P15	0	R/W	
		D4	**P14MUX**	P14 port function select	1	SDA	0	P14	0	R/W	
		D3	**P13MUX**	P13 port function select	1	FOUT1	0	P13	0	R/W	
		D2–0	**–**	reserved	–				–	–	0 when being read.

The P13–P15 and P17 I/O port pins are shared with a peripheral module. This register configures the pin functions.

D7 P17MUX: P17 Port Function Select Bit

1 (R/W): #SPISS (SPI)

0 (R/W): P17 port (default)

D6 Reserved

D5 P15MUX: P15 Port Function Select Bit

1 (R/W): SCL (I2C)

0 (R/W): P15 port (default)

D4 P14MUX: P14 Port Function Select Bit

1 (R/W): SDA (I2C)

0 (R/W): P14 port (default)

D3 P13MUX: P13 Port Function Select Bit

1 (R/W): FOUT1 (OSC)

0 (R/W): P13 port (default)

D[2:0] Reserved

0x52a2: P2 Port Function Select Register (P2_PMUX)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P2 Port Function Select Register (P2_PMUX)	**0x52a2** (8 bits)	D7	**P27MUX**	P27 port function select	1	EXCL3	0	P27	0	R/W	
		D6	**P26MUX**	P26 port function select	1	TOUT	0	P26	0	R/W	
		D5	**P25MUX**	P25 port function select	1	SCLK	0	P25	0	R/W	
		D4	**P24MUX**	P24 port function select	1	SOUT	0	P24	0	R/W	
		D3	**P23MUX**	P23 port function select	1	SIN	0	P23	0	R/W	
		D2	**P22MUX**	P22 port function select	1	SPICLK	0	P22	0	R/W	
		D1	**P21MUX**	P21 port function select	1	SDO	0	P21	0	R/W	
		D0	**P20MUX**	P20 port function select	1	SDI	0	P20	0	R/W	

The P20–P27 I/O port pins are shared with a peripheral module. This register configures the pin functions.

D7 P27MUX: P27 Port Function Select Bit

1 (R/W): EXCL3 (T16E)
0 (R/W): P27 port (default)

D6 P26MUX: P26 Port Function Select Bit

1 (R/W): TOUT (T16E)
0 (R/W): P26 port (default)

D5 P25MUX: P25 Port Function Select Bit

1 (R/W): SCLK (UART)
0 (R/W): P25 port (default)

D4 P24MUX: P24 Port Function Select Bit

1 (R/W): SOUT (UART)
0 (R/W): P24 port (default)

D3 P23MUX: P23 Port Function Select Bit

1 (R/W): SIN (UART)
0 (R/W): P23 port (default)

D2 P22MUX: P22 Port Function Select Bit

1 (R/W): SPICLK (SPI)
0 (R/W): P22 port (default)

D1 P21MUX: P21 Port Function Select Bit

1 (R/W): SDO (SPI)
0 (R/W): P21 port (default)

D0 P20MUX: P20 Port Function Select Bit

1 (R/W): SDI (SPI)
0 (R/W): P20 port (default)

0x52a3: P3 Port Function Select Register (P3_PMUX)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P3 Port Function Select Register (P3_PMUX)	**0x52a3** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**P33MUX**	P33 port function select	1	P33	0	DSIO	0	R/W	
		D2	**P32MUX**	P32 port function select	1	P32	0	DST2	0	R/W	
		D1	**P31MUX**	P31 port function select	1	P31	0	DCLK	0	R/W	
		D0	**P30MUX**	P30 port function select	1	FOUT3	0	P30	0	R/W	

The P30–P33 I/O port pins are shared with a peripheral module. This register configures the pin functions.

D[7:4] Reserved

D3 P33MUX: P33 Port Function Select Bit

1 (R/W): P33 port

0 (R/W): DSIO (DBG) (default)

D2 P32MUX: P32 Port Function Select Bit

1 (R/W): P32 port

0 (R/W): DST2 (DBG) (default)

D1 P31MUX: P31 Port Function Select Bit

1 (R/W): P31 port

0 (R/W): DCLK (DBG) (default)

D0 P30MUX: P30 Port Function Select Bit

1 (R/W): FOUT3 (OSC)

0 (R/W): P30 port (default)

10.9 Precautions

Operating clock

- Supply PCLK from the clock generator before the I/O ports can be accessed.
  Furthermore, a prescaler output clock is required for operating the P0 port chattering filter. Turn the prescaler on when using the chattering filter.

Pull-up resistor

- When changing the port pin from low level to high level with the built-in pull-up resistor, a delay in the waveform rise time will occur depending on the time constant of the pull-up resistor and the load capacitance of the pin. Therefore it is necessary to set an appropriate wait time for reading an I/O port. This wait time should be the amount of time or more calculated by the following expression.

  Wait time = R_{IN} × (C_{IN} + load capacitance on the board) × 1.6 [seconds]

  R_{IN}: Pull-up resistance Max. value

  C_{IN}: Pin capacitance Max. value

- For unused ports, enable the pull-up resistors.

P0 and P1 port interrupt

- To avoid occurrence of unnecessary interrupts, be sure to reset the interrupt flags P0IF[7:0] (0x5207) or P1IF[7:0] (0x5217) corresponding to the ports used to generate an interrupt before the interrupt is enabled using the P0_IMSK register (0x5205) or P1_IMSK register (0x5215).
- Set the P0 and P1 interrupt trigger mode in the ITC to level trigger.
  After an interrupt occurs, reset the interrupt flag in the P0IF[7:0] (0x5207) or P1IF[7:0] (0x5217) of the P port module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

P0 port chattering filter

- Input interrupts cannot be accepted in SLEEP mode if the CPU enters SLEEP mode when the chattering filter is active. The chattering filter should be disabled (off) before executing the `slp` instruction.
- Be sure to disable the P0 port interrupt before changing the P0_CHAT register (0x5208). Unnecessary interrupt may occur if the register is changed when the P0 port interrupt has been enabled.
- The check time to eliminate chattering means the maximum pulse width that can be eliminated. The valid interrupt input needs a pulse width of the set check time (minimum) to twice that of the check time (maximum).
- The internal signal may oscillate if the rise/fall time of the input signal is too long because the input signal level transition to the threshold level duration of time is too long. This causes the input interrupt to malfunction, therefore setup the input signal so that the rise/fall time is 25 ns or less.

P0 port key-entry reset

- When using the P0 port key-entry reset function, make sure that the designated input ports will not be simultaneously set to low level while the application program is running.
- The P0 port key-entry reset function cannot be used for power-on reset as it must be enabled with software.
- The P0 port key-entry reset function cannot be used in SLEEP mode.

THIS PAGE IS BLANK.

11 16-bit Timers (T16)

11.1 Outline of the 16-bit Timers

The S1C17701 is equipped with three channels of 16-bit timers (T16).

The 16-bit timer includes a 16-bit presettable down counter and a 16-bit reload data register for setting the preset value. The timer counts down from the initial value set in the reload data register and outputs an underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and an internal serial interface clock. The underflow period can be programmed by selecting a prescaler clock and setting reload data. This allows the application program to get any desired time intervals and programmable serial transfer rates.

Furthermore, the 16-bit timers also have an event counter function and a pulse width measurement function using I/O port pins.

Figure 11.1.1 shows the structure of the 16-bit timer.

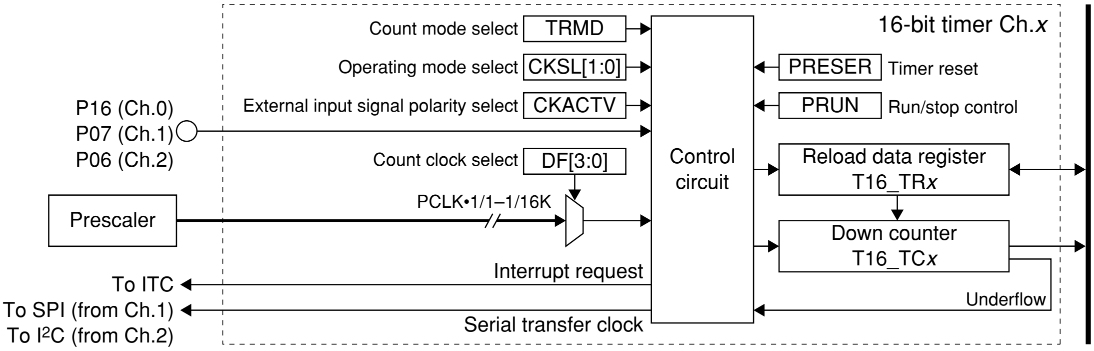


Figure 11.1.1 Structure of 16-bit Timer (one channel)

Note: The descriptions in this section apply to all 16-bit timer channels because the three channels of 16-bit timers have the same functions except for the control register addresses. The '*x*' in the register names denotes a channel number (0 to 2) and the register addresses are described as (*Ch.0*/*Ch.1*/*Ch.2*).

Example: T16_CTL*x* register (0x4226/0x4246/0x4266)
Ch.0: T16_CTL0 register (0x4226)
Ch.1: T16_CTL1 register (0x4246)
Ch.2: T16_CTL2 register (0x4266)

11.2 16-bit Timer Operating Mode

The 16-bit timer has three operating modes.
1. Internal clock mode (general timer to count the internal clock)
2. External clock mode (functions as an event counter)
3. Pulse width measurement mode (measures external input pulse widths by counting the internal clock)

Use CKSL[1:0] (D[9:8]/T16_CTL*x* register) to select the operating mode.

∗ **CKSL[1:0]**: Input Clock and Pulse Width Count Mode Select Bits in the 16-bit Timer Ch.*x* Control (T16_CTL*x*) Register (D[9:8]/0x4226/0x4246/0x4266)

Table 11.2.1 Selecting Operating Mode

CKSL[1:0]	Operating mode
0x3	Reserved
0x2	Pulse width measurement mode
0x1	External clock mode
0x0	Internal clock mode

(Default: 0x0)

11.2.1 Internal Clock Mode

In internal clock mode, the timer uses the prescaler output clock as the count clock.
The timer starts counting down from the counter initial value set in the reload data register and outputs the underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and a clock for the internal serial interface. The period until an underflow occurs can be programmed minutely according to the prescaler clock and counter initial value selections, so this mode is useful for generating a serial transfer clock or a one-shot time measurement.

Selecting the count clock

Use the DF[3:0] bits (D[3:0]/T16_CLK*x* register) to select the count clock from the 15 clocks, PCLK divided by 1 to PCLK divided by 16K, generated by the prescaler.

∗ **DF[3:0]**: Timer Input Clock Select Bits in the 16-bit Timer Ch.*x* Input Clock Select (T16_CLK*x*) Register (D[3:0]/0x4220/0x4240/0x4260)

Table 11.2.1.1 Selecting the Count Clock

DF[3:0]	Prescaler output clock	DF[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK•1/128
0xe	PCLK•1/16384	0x6	PCLK•1/64
0xd	PCLK•1/8192	0x5	PCLK•1/32
0xc	PCLK•1/4096	0x4	PCLK•1/16
0xb	PCLK•1/2048	0x3	PCLK•1/8
0xa	PCLK•1/1024	0x2	PCLK•1/4
0x9	PCLK•1/512	0x1	PCLK•1/2
0x8	PCLK•1/256	0x0	PCLK•1/1

(Default: 0x0)

Notes:
- Before the 16-bit timer can start counting in internal clock mode, the prescaler must be run.
- When setting the count clock, make sure the 16-bit timer counter is stopped.

For controlling the prescaler, see Chapter 9, "Prescaler (PSC)."

11.2.2 External Clock Mode

In external clock mode, the timer uses the clock or pulses input from an I/O port as the count clock. Thus the timer can be used as an event counter. The timer operation is the same as internal clock mode except the count clock source.

External clock input ports

The table below lists the external clock/pulse input ports.

Table 11.2.2.1 External Clock Input Ports

Timer channel	Input signal name	I/O port pin
Ch.0	EXCL0	P16
Ch.1	EXCL1	P07
Ch.2	EXCL2	P06

Make sure that the I/O port used for an external clock/pulse input is set to input mode (default). It is not necessary to select the pin function. Although the I/O port functions as a general-purpose input port, the input signal is also sent to the 16-bit timer.

The P07 and P06 ports that are respectively used by the 16-bit timer Ch.1 and Ch.2 incorporate a chattering filter and it is effective for the EXCL*x* input signal. For controlling the chattering filter, see Section 10.6, "Chattering Filter for P0 Ports."

Selecting the signal polarity

The external clock mode allows selection of a input signal edge to perform counting. Either falling edge or rising edge can be selected using the CKACTV (D10/T16_CTL*x* register).

* **CKACTV**: External Clock Active Level Select Bit in the 16-bit Timer Ch.*x* Control (T16_CTL*x*) Register (D10/0x4226/0x4246/0x4266)

When CKACTV is 1 (default), the timer counts down at the rising edge of the input signal; when it is set to 0, the timer counts down at the falling edge.

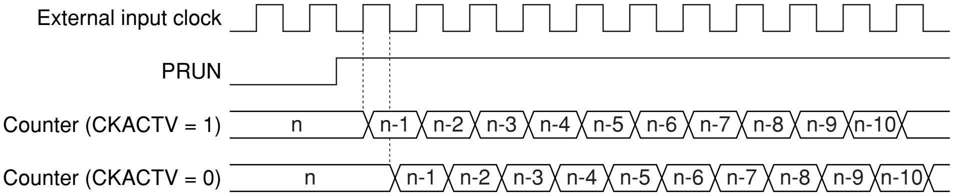


Figure 11.2.2.1 Count Timing in External Clock Mode

The 16-bit timer set in this mode does not use the prescaler. If the prescaler clocks are not used in other peripheral modules, the prescaler can be stopped to reduce current consumption. (Note that the P0 port chattering filter uses a prescaler clock.)

11.2.3 Pulse Width Measurement Mode

In pulse width measurement mode, the internal clock is supplied to the counter only while an external pulse input from the external clock port is at the active level specified. This makes it possible to generate an interrupt when a pulse longer than a specified width is input or to measure the input pulse width.

Pulse input port

The I/O ports used to input external pulses are the same as the external clock mode (see Table 11.2.2.1).

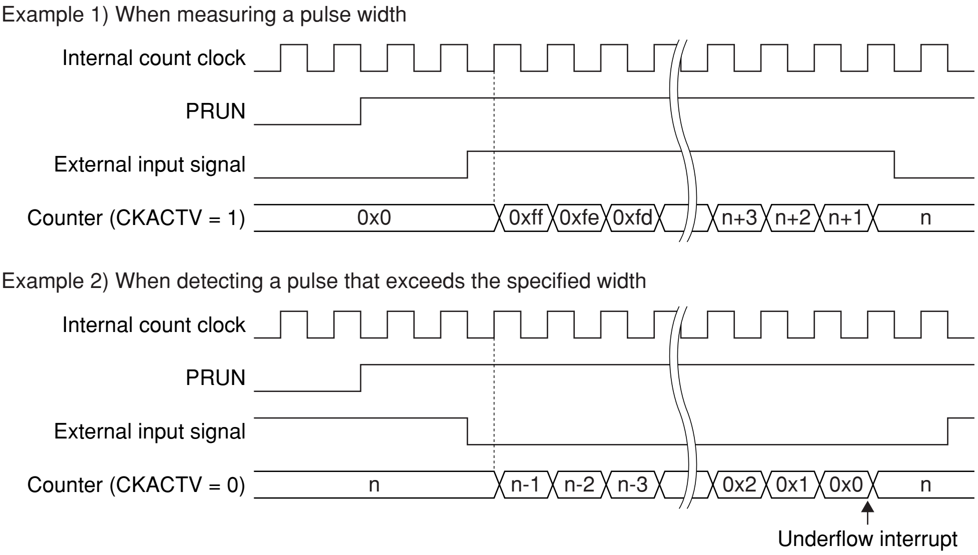
Set the I/O port, which corresponds to the timer channel used, to input mode and input pulses to be measured from the I/O pin.

Selecting the count clock

As in the case of the internal clock mode, the timer operates with the prescaler output clock selected using DF[3:0] (D[3:0]/T16_CLK*x* register). Select an appropriate clock according to the approximate input pulse width and measurement accuracy (see Table 11.2.1.1).

Selecting the signal polarity

Use CKACTV (D10/T16_CTL*x* register) to select the active level for the pulse to be measured. When CKACTV is 1 (default), the high period of the input pulse will be measured; when it is set to 0, the low period will be measured.



Example 1) When measuring a pulse width

Example 2) When detecting a pulse that exceeds the specified width

Figure 11.2.3.1 Count Operation in Pulse Width Measurement Mode

11.3 Count Mode

The 16-bit timer has two count modes: repeat mode and one-shot mode. It can be selected using the TRMD bit (D4/T16_CTL*x* register).

∗ **TRMD**: Count Mode Select Bit in the 16-bit Timer Ch.*x* Control (T16_CTL*x*) Register (D4/0x4226/0x4246/0x4266)

Repeat mode (TRMD = 0, default)

The 16-bit timer is set in repeat mode when TRMD is set to 0.

In this mode, the 16-bit timer does not stop after it starts counting until the application program stops the timer. When the counter underflows, the timer presets the reload data register value to the counter and continues counting. The timer outputs the underflow pulses periodically. Set the 16-bit timer in this mode when generating periodical interrupts with a given interval or generating the serial transfer clock.

One-shot mode (TRMD = 1)

The 16-bit timer is set in one-shot mode when TRMD is set to 1.

In this mode, the 16-bit timer automatically stops counting when the counter underflows, so only one interrupt can be generated after starting the timer. When an underflow occurs, the counter is preset with the reload data register value before the timer operation stops. Set the 16-bit timer in this mode when a certain waiting time must be generated or measuring pulse widths.

11.4 16-bit Timer Reload Register and Underflow Period

The Reload Data (T16_TR*x*) Register (0x4222/0x4242/0x4262) is used to set the initial value to the down counter. The counter initial value set in the reload data register is preset to the down counter when the 16-bit timer is reset or when the counter underflows. When starting the 16-bit timer after resetting, the timer counts down from the reload value. So the reload value and the input clock frequency determine the period of time from starting the timer until an underflow occurs (and between underflows). This makes it possible to obtain a desired wait time, a periodical interrupt interval, or programmable transfer clock for the serial interface.

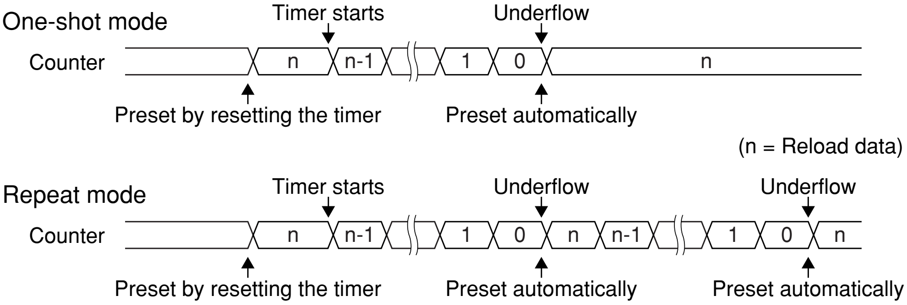


Figure 11.4.1 Preset Timing

The underflow period is calculated by the expression below.

$$\text{Underflow period} = \frac{\text{TR} + 1}{\text{clk\_in}} \ [\text{s}] \qquad \text{Underflow cycle} = \frac{\text{clk\_in}}{\text{TR} + 1} \ [\text{Hz}]$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
TR: Reload data (0–65535)

11.5 Resetting the 16-bit Timer

To reset the 16-bit timer, write 1 to the PRESER bit (D1/T16_CTL*x* register). This initializes the counter by presetting the reload data register value.

∗ **PRESER**: Timer Reset Bit in the 16-bit Timer Ch.*x* Control (T16_CTL*x*) Register (D1/0x4226/0x4246/0x4266)

11.6 16-bit Timer Run/Stop Control

Before starting the 16-bit timer, set up the conditions as shown below.

(1) Select an operating mode (internal clock, external clock, pulse width measurement). See Section 11.2.
(2) In internal clock mode or pulse width measurement mode, select the count clock (prescaler output clock). See Section 11.2.1.
(3) Select a count mode (one-shot or repeat). See Section 11.3.
(4) Calculate the counter initial value and set it to the reload data register. See Section 11.4.
(5) Reset the timer to preset the initial value to the counter. See Section 11.5.
(6) Set up the interrupt level and enable the interrupt of the timer channel if the timer interrupt is used. See Section 11.8.

To start the 16-bit timer, write 1 to the PRUN bit (D0/T16_CTL*x* register).

* **PRUN**: Timer Run/Stop Control Bit in the 16-bit Timer Ch.*x* Control (T16_CTL*x*) Register (D0/0x4226/0x4246/0x4266)

The timer starts counting down from the initial value or the current counter value if the initial value has not been preset. When the counter underflows, the timer outputs an underflow pulse and presets the initial value again. At the same time, an interrupt request is sent to the interrupt controller (ITC).
If the timer is set in one-shot mode, the timer stops counting.
If the timer is set in repeat mode, the timer continues counting from the reloaded initial value.

To stop the 16-bit timer from the application program, write 0 to the PRUN bit. The counter stops counting and holds the current counter value until the timer is reset or restarted. To restart counting from the initial value, reset the timer before writing 1 to the PRUN bit.

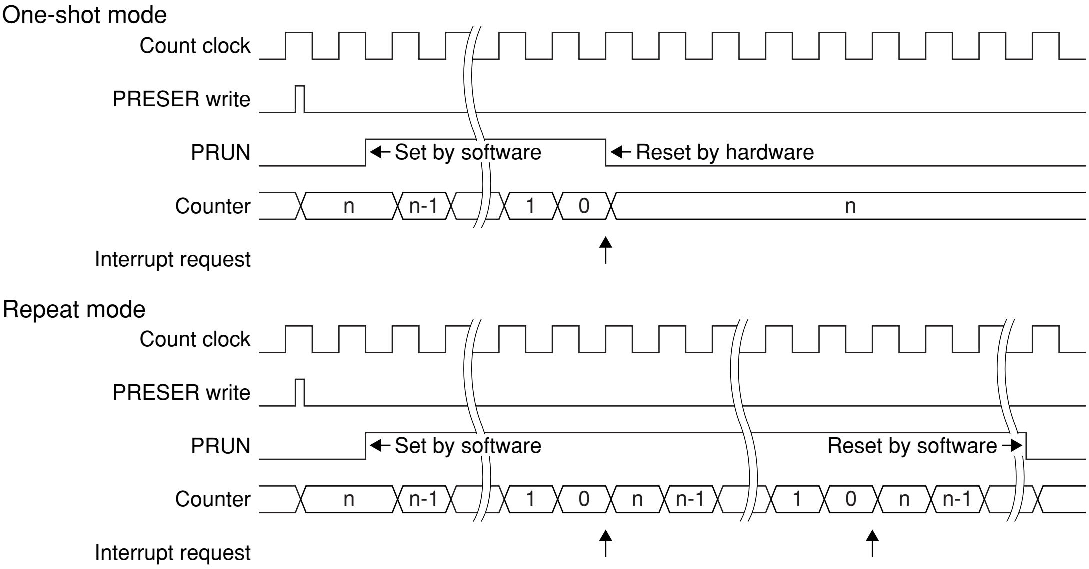


Figure 11.6.1 Count Operation

In pulse width measurement mode, the timer performs counting only when PRUN = 1 and the external input signal is the specified active level. When the external input signal goes inactive, the 16-bit timer stops counting and maintains the count value until the next active pulse is input. (See Figure 11.2.3.1.)

11.7 16-bit Timer Output Signal

The 16-bit timer outputs an underflow pulse when the counter underflows.
This pulse is used to request a timer interrupt.
Also this pulse is used to generate a serial transfer clock for the internal serial interface.

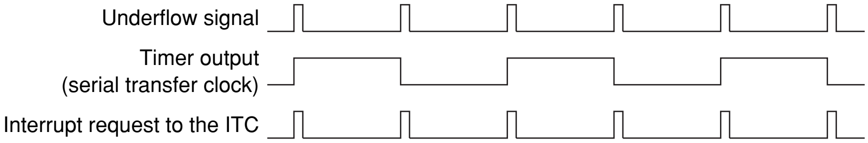


Figure 11.7.1 Timer Output Clock

The generated clocks are sent to the internal serial interfaces as below.
16-bit timer Ch.1 output clock → SPI
16-bit timer Ch.2 output clock → I^2C

The reload data register value to obtain a desired transfer rate is calculated by the expression below.

SPI $$TR = \frac{clk\_in}{bps \times 2} - 1$$

I^2C $$TR = \frac{clk\_in}{bps \times 2} - 1$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
TR: Reload data (0–65535)
bps: Transfer rate (bits/second)

11.8 16-bit Timer Interrupt

The 16-bit timer outputs an interrupt request signal to the interrupt controller (ITC) when the counter underflows. To generate a timer underflow interrupt, set up the interrupt level and enable the interrupt using the ITC registers.

ITC registers for timer interrupts

Table 11.8.1 shows the control registers of the ITC provided for each timer channel.

Table 11.8.1 ITC Registers

Timer channel	Interrupt flag	Interrupt enable bit	Interrupt level setup bits
Ch.0	IIFT1 (D9/ITC_IFLG)	IIEN1 (D9/ITC_EN)	IILV1[2:0] (D[10:8]/ITC_ILV0)
Ch.1	IIFT2 (D10/ITC_IFLG)	IIEN2 (D10/ITC_EN)	IILV2[2:0] (D[2:0]/ITC_ILV1)
Ch.2	IIFT3 (D11/ITC_IFLG)	IIEN3 (D11/ITC_EN)	IILV3[2:0] (D[10:8]/ITC_ILV1)

ITC_IFLG register (0x4300)
ITC_EN register (0x4302)
ITC_ILV0 register (0x430e)
ITC_ILV1 register (0x4310)

When an underflow occurs in the timer, the corresponding interrupt flag is set to 1.

If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the timer interrupt, set the interrupt enable bit to 0.

The interrupt flag is always set to 1 by the timer underflow pulse, regardless of how the interrupt enable bit is set (even when set to 0).

The interrupt level setup bits set the interrupt level (0 to 7) of the timer interrupt. If the same interrupt level is set, timer Ch.0 has highest priority and timer Ch.2 has lowest priority.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The timer interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Chapter 6, “Interrupt Controller (ITC).”

Interrupt vectors

The following shows the vector numbers and vector addresses for the timer interrupt:

Table 11.8.2 Timer Interrupt Vectors

Timer channel	Vector number	Vector address
Timer Ch.0	13 (0x0d)	0x8034
Timer Ch.1	14 (0x0e)	0x8038
Timer Ch.2	15 (0x0f)	0x803c

11.9 Details of Control Registers

Table 11.9.1 List of 16-bit Timer Registers

Address	Register name		Function
0x4220	T16_CLK0	16-bit Timer Ch.0 Input Clock Select Register	Selects a prescaler output clock.
0x4222	T16_TR0	16-bit Timer Ch.0 Reload Data Register	Sets reload data.
0x4224	T16_TC0	16-bit Timer Ch.0 Counter Data Register	Counter data
0x4226	T16_CTL0	16-bit Timer Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
0x4240	T16_CLK1	16-bit Timer Ch.1 Input Clock Select Register	Selects a prescaler output clock.
0x4242	T16_TR1	16-bit Timer Ch.1 Reload Data Register	Sets reload data.
0x4244	T16_TC1	16-bit Timer Ch.1 Counter Data Register	Counter data
0x4246	T16_CTL1	16-bit Timer Ch.1 Control Register	Sets the timer mode and starts/stops the timer.
0x4260	T16_CLK2	16-bit Timer Ch.2 Input Clock Select Register	Selects a prescaler output clock.
0x4262	T16_TR2	16-bit Timer Ch.2 Reload Data Register	Sets reload data.
0x4264	T16_TC2	16-bit Timer Ch.2 Counter Data Register	Counter data
0x4266	T16_CTL2	16-bit Timer Ch.2 Control Register	Sets the timer mode and starts/stops the timer.

The following describes each 16-bit timer register. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4220/0x4240/0x4260: 16-bit Timer Ch.*x* Input Clock Select Registers (T16_CLK*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
16-bit Timer Ch.*x* Input Clock Select Register (T16_CLK*x*)	**0x4220 0x4240 0x4260** (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			

Note: The letter '*x*' in register names, etc., denotes a channel number from 0 to 2.

0x4220: 16-bit Timer Ch.0 Input Clock Select Register (T16_CLK0)
0x4240: 16-bit Timer Ch.1 Input Clock Select Register (T16_CLK1)
0x4260: 16-bit Timer Ch.2 Input Clock Select Register (T16_CLK2)

D[15:4] Reserved

D[3:0] DF[3:0]: Timer Input Clock Select Bits

These bits select the count clock of the 16-bit timer from 15 prescaler output clocks.

Table 11.9.2 Selecting the Count Clock

DF[3:0]	Prescaler output clock	DF[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK•1/128
0xe	PCLK•1/16384	0x6	PCLK•1/64
0xd	PCLK•1/8192	0x5	PCLK•1/32
0xc	PCLK•1/4096	0x4	PCLK•1/16
0xb	PCLK•1/2048	0x3	PCLK•1/8
0xa	PCLK•1/1024	0x2	PCLK•1/4
0x9	PCLK•1/512	0x1	PCLK•1/2
0x8	PCLK•1/256	0x0	PCLK•1/1

(Default: 0x0)

Note: When setting the count clock, make sure the 16-bit timer counter is stopped.

0x4222/0x4242/0x4262: 16-bit Timer Ch.*x* Reload Data Registers (T16_TR*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.*x* Reload Data Register (T16_TR*x*)	**0x4222** **0x4242** **0x4262** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	

Note: The letter '*x*' in register names, etc., denotes a channel number from 0 to 2.

0x4222: 16-bit Timer Ch.0 Reload Data Register (T16_TR0)
0x4242: 16-bit Timer Ch.1 Reload Data Register (T16_TR1)
0x4262: 16-bit Timer Ch.2 Reload Data Register (T16_TR2)

D[15:0] TR[15:0]: 16-bit Timer Reload Data

Set the initial value for the counter. (Default: 0x0)

The reload data written in this register is preset to the respective counter when the timer is reset or when the counter underflows.

When starting the 16-bit timer after resetting, the timer counts down from the reload value. So the reload value and the input clock frequency determine the period of time from starting the timer until an underflow occurs (and between underflows). This makes it possible to obtain a desired wait time, a periodical interrupt interval, or programmable transfer clock for the serial interface.

0x4224/0x4244/0x4264: 16-bit Timer Ch.*x* Counter Data Registers (T16_TC*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.*x* Counter Data Register (T16_TC*x*)	**0x4224** **0x4244** **0x4264** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	

Note: The letter '*x*' in register names, etc., denotes a channel number from 0 to 2.

0x4224: 16-bit Timer Ch.0 Counter Data Register (T16_TC0)
0x4244: 16-bit Timer Ch.1 Counter Data Register (T16_TC1)
0x4264: 16-bit Timer Ch.2 Counter Data Register (T16_TC2)

D[15:0] TC[15:0]: 16-bit Timer Counter Data

The counter data can be read from this register. (Default: 0xffff)
This is a read-only register, so the writing operation is invalid.

0x4226/0x4246/0x4266: 16-bit Timer Ch.*x* Control Registers (T16_CTL*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
16-bit Timer Ch.*x* Control Register (T16_CTL*x*)	**0x4226** **0x4246** **0x4266** (16 bits)	D15–11	–	reserved	–		–	–	0 when being read.
		D10	**CKACTV**	External clock active level select	1 High	0 Low	1	R/W	
		D9–8	**CKSL[1:0]**	Input clock and pulse width measurement mode select	CKSL[1:0]	Mode	0x0	R/W	
					0x3	reserved			
					0x2	Pulse width			
					0x1	External clock			
					0x0	Internal clock			
		D7–5	–	reserved	–		–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot	0 Repeat	0	R/W	
		D3–2	–	reserved	–		–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset	0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run	0 Stop	0	R/W	

Note: The letter '*x*' in register names, etc., denotes a channel number from 0 to 2.

0x4226: 16-bit Timer Ch.0 Control Register (T16_CTL0)
0x4246: 16-bit Timer Ch.1 Control Register (T16_CTL1)
0x4266: 16-bit Timer Ch.2 Control Register (T16_CTL2)

D[15:11] Reserved

D10 CKACTV: External Clock Active Level Select Bit

Selects an external input pulse polarity or a count edge of the external clock.

1 (R/W): Active high/rising edge (default)
0 (R/W): Active low/falling edge

In external clock mode (CKSL[1:0] = 0x1), select either rising edges of the external input clock or falling edges as the count timings. In pulse width measurement mode (CKSL[1:0] = 0x2), select the polarity of the external input pulse.

D[9:8] CKSL[1:0]: Input Clock and Pulse Width Measurement Mode Select Bits

Selects a 16-bit timer operating mode.

Table 11.9.3 Selecting Operating Mode

CKSL[1:0]	Operating mode
0x3	Reserved
0x2	Pulse width measurement mode
0x1	External clock mode
0x0	Internal clock mode

(Default: 0x0)

In internal clock mode, the timer uses the prescaler output clock as the count clock. The timer starts counting down from the counter initial value set in the reload data register and outputs the underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and a clock for the internal serial interface. The period until an underflow occurs can be programmed minutely according to the prescaler clock and counter initial value selections, so this mode is useful for generating a serial transfer clock or a one-shot time measurement.

In external clock mode, the timer uses the clock or pulses input from an I/O port (Ch.0: P16, Ch.1: P07, Ch.2: P06) as the count clock. Thus the timer can be used as an event counter. The timer operation is the same as internal clock mode except the count clock source.

In pulse width measurement mode, the internal clock is supplied to the counter only while an external pulse input from the external clock port is the active level specified. This makes it possible to generate an interrupt when a pulse longer than a specified width is input or to measure the input pulse width.

D[7:5] Reserved

D4 TRMD: Count Mode Select Bit

Selects the count mode of the 16-bit timer.
1 (R/W): One-shot mode
0 (R/W): Repeat mode (default)

The 16-bit timer is set in repeat mode when TRMD is set to 0. In this mode, the 16-bit timer does not stop after it starts counting until the application program stops the timer. When the counter underflows, the timer presets the reload data register value to the counter and continues counting. The timer outputs the underflow pulses periodically. Set the 16-bit timer in this mode when generating periodical interrupts with a given interval or generating the serial transfer clock.

The 16-bit timer is set in one-shot mode when TRMD is set to 1. In this mode, the 16-bit timer automatically stops counting when the counter underflows, so only one interrupt can be generated after starting the timer. When an underflow occurs, the counter is preset with the reload data register value before the timer operation stops. Set the 16-bit timer in this mode when a certain waiting time must be generated or measuring pulse widths.

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

Resets the 16-bit timer.
1 (W): Reset
0 (W): Has no effect
0 (R): Always 0 when read (default)

Writing 1 to this bit presets the reload data in the counter.

D0 PRUN: Timer Run/Stop Control Bit

Controls the timer's Run/Stop state.
1 (R/W): Run
0 (R/W): Stop (default)

The timer starts counting by writing 1 to PRUN and stops counting by writing 0.
In the stop state, the counter data is retained until the timer is reset or placed in a run state.

11.10 Precautions

- Before the 16-bit timer can start counting, the prescaler must be run.
- When setting the count clock or count mode, make sure the 16-bit timer is turned off.

THIS PAGE IS BLANK.

12 8-bit Timer (T8F)

12.1 Outline of the 8-bit Timer

The S1C17701 incorporates one channel of 8-bit timer with fine mode.

The 8-bit timer module includes an 8-bit presettable down counter and an 8-bit reload data register for setting the preset value. The timer counts down from the initial value set in the reload data register and outputs an underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and the clock for UART. The underflow period can be programmed by selecting a prescaler clock and setting reload data. This allows the application program to get any desired time intervals and programmable serial transfer rates. The fine mode provides a function to minimize transfer rate error.

Figure 12.1.1 shows the structure of the 8-bit timer.

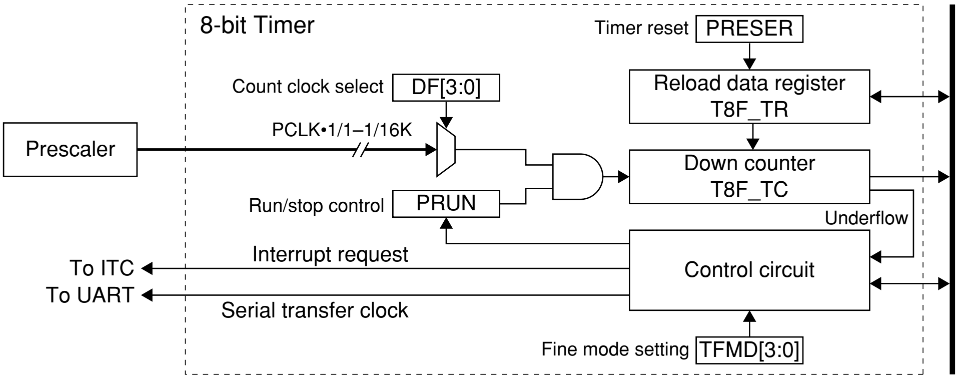


Figure 12.1.1 Structure of 8-bit Timer

12.2 Count Mode of the 8-bit Timer

The 8-bit timer has two count modes: repeat mode and one-shot mode. It can be selected using the TRMD bit (D4/ T8F_CTL register).

∗ **TRMD**: Count Mode Select Bit in the 8-bit Timer Control (T8F_CTL) Register (D4/0x4206)

Repeat mode (TRMD = 0, default)

The 8-bit timer is set in repeat mode when TRMD is set to 0.
In this mode, the 8-bit timer does not stop after it starts counting until the application program stops the timer. When the counter underflows, the timer presets the reload data register value to the counter and continues counting. The timer outputs the underflow pulses periodically. Set the 8-bit timer in this mode when generating periodical interrupts with a given interval or generating the serial transfer clock.

One-shot mode (TRMD = 1)

The 8-bit timer is set in one-shot mode when TRMD is set to 1.
In this mode, the 8-bit timer automatically stops counting when the counter underflows, so only one interrupt can be generated after starting the timer. When an underflow occurs, the counter is preset with the reload data register value before the timer operation stops. Set the 8-bit timer in this mode when a certain waiting time must be generated.

Note: When setting the count mode, make sure the 8-bit timer counter is stopped.

12.3 Count Clock

The 8-bit timer uses a prescaler output clock as the count clock. The prescaler divides the PCLK clock by 1 to 16K to generate 15 clocks. Select one of the prescaler output clocks using the DF[3:0] bits (D[3:0]/T8F_CLK register).

∗ **DF[3:0]**: Timer Input Clock Select Bits in the 8-bit Timer Input Clock Select (T8F_CLK) Register (D[3:0]/0x4200)

Table 12.3.1 Selecting the Count Clock

DF[3:0]	Prescaler output clock	DF[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK•1/128
0xe	PCLK•1/16384	0x6	PCLK•1/64
0xd	PCLK•1/8192	0x5	PCLK•1/32
0xc	PCLK•1/4096	0x4	PCLK•1/16
0xb	PCLK•1/2048	0x3	PCLK•1/8
0xa	PCLK•1/1024	0x2	PCLK•1/4
0x9	PCLK•1/512	0x1	PCLK•1/2
0x8	PCLK•1/256	0x0	PCLK•1/1

(Default: 0x0)

Notes:
- Before the 8-bit timer can start counting, the prescaler must be run.
- When setting the count clock, make sure the 8-bit timer counter is stopped.

For controlling the prescaler, see Chapter 9, “Prescaler (PSC).”

12.4 8-bit Timer Reload Register and Underflow Period

The Reload Data (T8F_TR) Register (0x4202) is used to set the initial value to the down counter.
The counter initial value set in the reload data register is preset to the down counter when the 8-bit timer is reset or when the counter underflows. When starting the 8-bit timer after resetting, the timer counts down from the reload value. So the reload value and the input clock frequency determine the period of time from starting the timer until an underflow occurs (and between underflows). This makes it possible to obtain a desired wait time, a periodical interrupt interval, or programmable transfer clock for the serial interface.

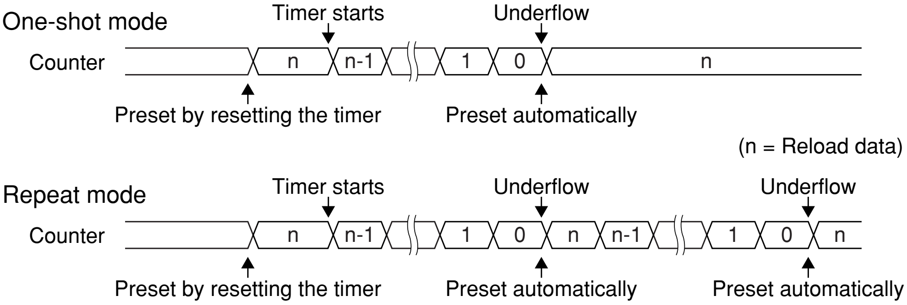


Figure 12.4.1 Preset Timing

The underflow period is calculated by the expression below.

$$\text{Underflow period} = \frac{\text{T8F\_TR} + 1}{\text{clk\_in}} \text{ [s]} \qquad \text{Underflow cycle} = \frac{\text{clk\_in}}{\text{T8F\_TR} + 1} \text{ [Hz]}$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
T8F_TR: Reload data (0–255)

Note: The UART divides the 8-bit timer output by 16 to generate the sampling clock. Make sure of the division ratio when setting a transfer rate.

12.5 Resetting the 8-bit Timer

To reset the 8-bit timer, write 1 to the PRESER bit (D1/T8F_CTL register). This initializes the counter by presetting the Reload Data Register value.

∗ **PRESER**: Timer Reset Bit in the 8-bit Timer Control (T8F_CTL) Register (D1/0x4206)

12.6 8-bit Timer Run/Stop Control

Before starting the 8-bit timer, set up the conditions as shown below.
(1) Select a count mode (one-shot or repeat). See Section 12.2.
(2) Select the count clock (prescaler output clock). See Section 12.3.
(3) Calculate the counter initial value and set it to the reload data register. See Section 12.4.
(4) Reset the timer to preset the initial value to the counter. See Section 12.5.
(5) Set up the interrupt level and enable the interrupt if the timer interrupt is used. See Section 12.9.

To start the 8-bit timer, write 1 to the PRUN bit (D0/T8F_CTL register).

∗ **PRUN**: Timer Run/Stop Control Bit in the 8-bit Timer Control (T8F_CTL) Register (D0/0x4206)

The timer starts counting down from the initial value or the current counter value if the initial value has not been preset. When the counter underflows, the timer outputs an underflow pulse and presets the initial value again. At the same time, an interrupt request is sent to the interrupt controller (ITC).
If the timer is set in one-shot mode, the timer stops counting.
If the timer is set in repeat mode, the timer continues counting from the reloaded initial value.

To stop the 8-bit timer from the application program, write 0 to the PRUN bit. The counter stops counting and holds the current counter value until the timer is reset or restarted. To restart counting from the initial value, reset the timer before writing 1 to the PRUN bit.

When the timer is reset during running, the timer loads the reload register value to the counter and continues counting.

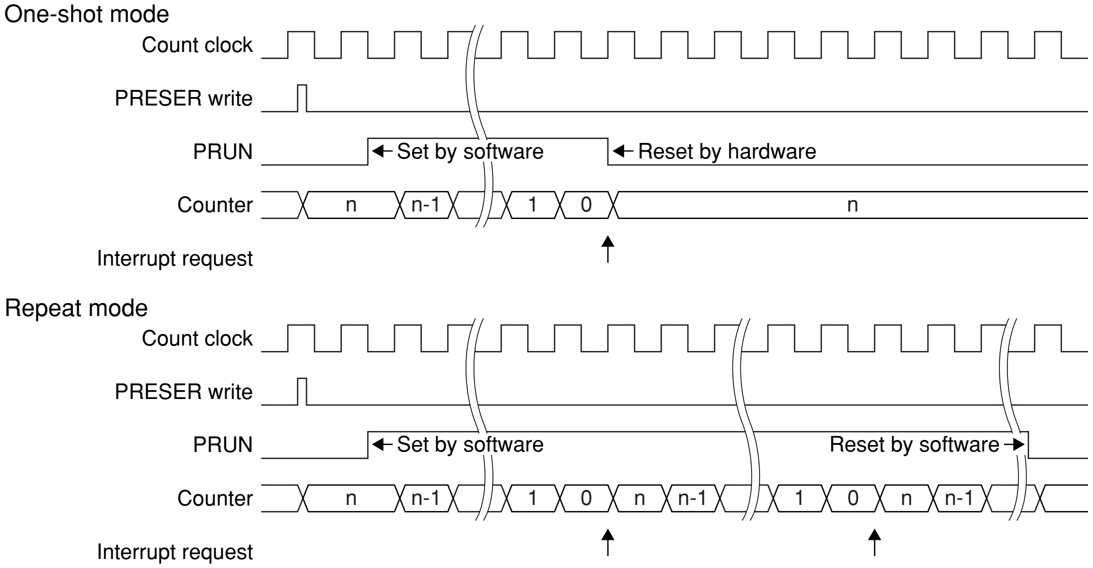


Figure 12.6.1 Count Operation

12.7 8-bit Timer Output Signal

The 8-bit timer outputs an underflow pulse when the counter underflows.
This pulse is used to request a timer interrupt.

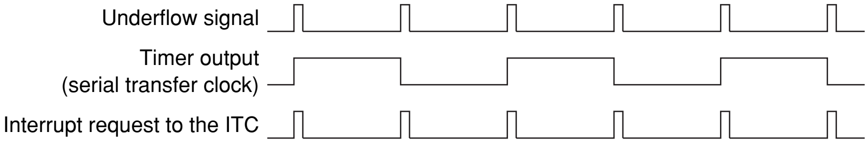


Figure 12.7.1 Timer Output Clock

Also this pulse is used to generate a serial transfer clock and the clock is sent to the UART.

The reload data register value to obtain a desired transfer rate is calculated by the expression below.

$$\text{bps} = \frac{\text{clk\_in}}{\{(\text{T8F\_TR} + 1) \times 16 + \text{TFMD}\}}$$

$$\text{T8F\_TR} = \left(\frac{\text{clk\_in}}{\text{bps}} - \text{TFMD} - 16\right) \div 16$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
T8F_TR: Reload data (0–255)
bps: Transfer rate (bits/second)
TFMD: Fine mode setting value (0–15)

12.8 Fine Mode

The fine mode provides a function to minimize transfer rate error.

The 8-bit timer can output a programmable clock used as the serial transfer clock for the UART. By selecting an appropriate prescaler output clock and reload data, the timer output clock can be configured with the desired frequency. However, an error may be introduced depending on the transfer rate. In fine mode, the counter delays outputting the underflow pulse to prolong the output clock period. The amount of delay can be specified using the TFMD[3:0] bits (D[11:8]/T8F_CTL register).

∗ **TFMD[3:0]**: Fine Mode Setup Bits in the 8-bit Timer Control (T8F_CTL) Register (D[11:8]/0x4206)

The TFMD[3:0] bits specify a pattern of delays to be inserted in a 16-underflow period. The output clock period will be prolonged for one count clock period per one delay inserted. Also this setting will delay interrupt timings.

Table 12.8.1 Delay Patterns Specified with TFMD[3:0]

TFMD[3:0]	Underflow number															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0x0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
0x1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	D
0x2	–	–	–	–	–	–	–	D	–	–	–	–	–	–	–	D
0x3	–	–	–	–	–	–	–	D	–	–	–	D	–	–	–	D
0x4	–	–	–	D	–	–	–	D	–	–	–	D	–	–	–	D
0x5	–	–	–	D	–	–	–	D	–	–	–	D	–	D	–	D
0x6	–	–	–	D	–	D	–	D	–	–	–	D	–	D	–	D
0x7	–	–	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x8	–	D	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x9	–	D	–	D	–	D	–	D	–	D	–	D	–	D	D	D
0xa	–	D	–	D	–	D	D	D	–	D	–	D	–	D	D	D
0xb	–	D	–	D	–	D	D	D	–	D	D	D	–	D	D	D
0xc	–	D	D	D	–	D	D	D	–	D	D	D	–	D	D	D
0xd	–	D	D	D	–	D	D	D	–	D	D	D	D	D	D	D
0xe	–	D	D	D	D	D	D	D	–	D	D	D	D	D	D	D
0xf	–	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

D: Indicates that a delay is inserted.

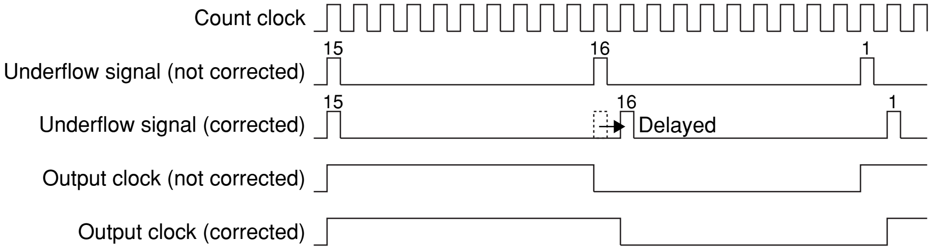

Count clock
Underflow signal (not corrected)
15
16
1
Underflow signal (corrected)
15
16
1
Delayed
Output clock (not corrected)
Output clock (corrected)

Figure 12.8.1 Delay Cycle Insertion in Fine Mode

At initial reset, TFMD[3:0] is set to 0x0. No delay will be inserted in this setting.

12.9 8-bit Timer Interrupt

The 8-bit timer outputs an interrupt request signal to the interrupt controller (ITC) when the counter underflows. To generate a timer underflow interrupt, set up the interrupt level and enable the interrupt using the ITC registers.

ITC registers for timer interrupts

The following shows the control bits of the ITC provided for the 8-bit timer:

Interrupt flag IIFT0

∗ **IIFT0**: 8-bit Timer Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D8/0x4300)

Interrupt enable bit IIEN0

∗ **IIEN0**: 8-bit Timer Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D8/0x4302)

Interrupt level setup bits IILV0

∗ **IILV0[2:0]**: 8-bit Timer Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV0) Register 0 (D[2:0]/0x430e)

When an underflow occurs in the timer, the corresponding interrupt flag is set to 1.
If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the timer interrupt, set the interrupt enable bit to 0.
The interrupt flag is always set to 1 by the timer underflow pulse, regardless of how the interrupt enable bit is set (even when set to 0).
The interrupt level setup bits set the interrupt level (0 to 7) of the timer interrupt.
An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The timer interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Chapter 6, “Interrupt Controller (ITC).”

Interrupt vector

The following shows the vector number and vector address for the timer interrupt:

Vector number: 12 (0x0c)
Vector address: 0x8030

12.10 Details of Control Registers

Table 12.10.1 List of 8-bit Timer Registers

Address	Register name		Function
0x4200	T8F_CLK	8-bit Timer Input Clock Select Register	Selects a prescaler output clock.
0x4202	T8F_TR	8-bit Timer Reload Data Register	Sets reload data.
0x4204	T8F_TC	8-bit Timer Counter Data Register	Counter data
0x4206	T8F_CTL	8-bit Timer Control Register	Sets the timer mode and starts/stops the timer.

The following describes each 8-bit timer register. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4200: 8-bit Timer Input Clock Select Register (T8F_CLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
8-bit Timer Input Clock Select Register (T8F_CLK)	**0x4200** (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			

D[15:4] Reserved

D[3:0] DF[3:0]: Timer Input Clock Select Bits

These bits select the count clock of the 8-bit timer from 15 prescaler output clocks.

Table 12.10.2 Selecting the Count Clock

DF[3:0]	Prescaler output clock	DF[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK•1/128
0xe	PCLK•1/16384	0x6	PCLK•1/64
0xd	PCLK•1/8192	0x5	PCLK•1/32
0xc	PCLK•1/4096	0x4	PCLK•1/16
0xb	PCLK•1/2048	0x3	PCLK•1/8
0xa	PCLK•1/1024	0x2	PCLK•1/4
0x9	PCLK•1/512	0x1	PCLK•1/2
0x8	PCLK•1/256	0x0	PCLK•1/1

(Default: 0x0)

Note: When setting the count clock, make sure the 8-bit timer counter is stopped.

0x4202: 8-bit Timer Reload Data Register (T8F_TR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Reload Data Register (T8F_TR)	**0x4202** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TR[7:0]**	8-bit timer reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	

D[15:8] Reserved

D[7:0] TR[7:0]: 8-bit Timer Reload Data

Set the initial value for the counter. (Default: 0x0)

The reload data written in this register is preset to the respective counter when the timer is reset or when the counter underflows.

When starting the 8-bit timer after resetting, the timer counts down from the reload value. So the reload value and the input clock frequency determine the period of time from starting the timer until an underflow occurs (and between underflows). This makes it possible to obtain a desired wait time, a periodical interrupt interval, or programmable transfer clock for the serial interface.

0x4204: 8-bit Timer Counter Data Register (T8F_TC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Counter Data Register (T8F_TC)	**0x4204** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TC[7:0]**	8-bit timer counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	

D[15:8] Reserved

D[7:0] TC[7:0]: 8-bit Timer Counter Data

The counter data can be read from this register. (Default: 0xff)

This is a read-only register, so the writing operation is invalid.

0x4206: 8-bit Timer Control Register (T8F_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit Timer Control Register (T8F_CTL)	**0x4206** (16 bits)	D15–12	–	reserved	–				–	–	0 when being read.
		D11–8	**TFMD[3:0]**	Fine mode setup	0x0 to 0xf				0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	–	reserved	–				–	–	0 when being read.
		D4	**TRMD**	Count mode select	1	One shot	0	Repeat	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	**PRESER**	Timer reset	1	Reset	0	Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1	Run	0	Stop	0	R/W	

D[15:12] Reserved

D[11:8] TFMD[3:0]: Fine Mode Setup Bits

Corrects transfer rate error. (Default: 0x0)

The TFMD[3:0] bits specify a pattern of delays to be inserted in a 16-underflow period. The output clock period will be prolonged for one count clock period per one delay inserted.

Table 12.10.3 Delay Patterns Specified with TFMD[3:0]

TFMD[3:0]	Underflow number															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0x0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
0x1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	D
0x2	–	–	–	–	–	–	–	D	–	–	–	–	–	–	–	D
0x3	–	–	–	–	–	–	–	D	–	–	–	D	–	–	–	D
0x4	–	–	–	D	–	–	–	D	–	–	–	D	–	–	–	D
0x5	–	–	–	D	–	–	–	D	–	–	–	D	–	D	–	D
0x6	–	–	–	D	–	D	–	D	–	–	–	D	–	D	–	D
0x7	–	–	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x8	–	D	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x9	–	D	–	D	–	D	–	D	–	D	–	D	–	D	D	D
0xa	–	D	–	D	–	D	D	D	–	D	–	D	–	D	D	D
0xb	–	D	–	D	–	D	D	D	–	D	D	D	–	D	D	D
0xc	–	D	D	D	–	D	D	D	–	D	D	D	–	D	D	D
0xd	–	D	D	D	–	D	D	D	–	D	D	D	D	D	D	D
0xe	–	D	D	D	D	D	D	D	–	D	D	D	D	D	D	D
0xf	–	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

D: Indicates that a delay is inserted.

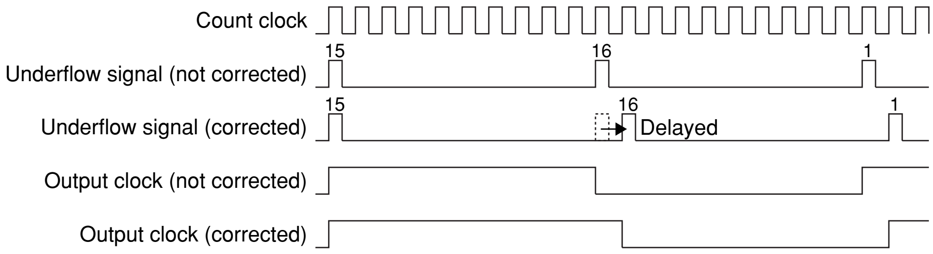

Count clock
15
16
1
Underflow signal (not corrected)
15
16
Delayed
1
Underflow signal (corrected)
Output clock (not corrected)
Output clock (corrected)

Figure 12.10.1 Delay Cycle Inserted in Fine Mode

D[7:5] Reserved

D4 TRMD: Count Mode Select Bit

Selects the count mode of the 8-bit timer.

1 (R/W): One-shot mode
0 (R/W): Repeat mode (default)

The 8-bit timer is set in repeat mode when TRMD is set to 0. In this mode, the 8-bit timer does not stop after it starts counting until the application program stops the timer. When the counter underflows, the timer presets the reload data register value to the counter and continues counting. The timer outputs the underflow pulses periodically. Set the 8-bit timer in this mode when generating periodical interrupts with a given interval or generating the serial transfer clock.

The 8-bit timer is set in one-shot mode when TRMD is set to 1. In this mode, the 8-bit timer automatically stops counting when the counter underflows, so only one interrupt can be generated after starting the timer. When an underflow occurs, the counter is preset with the reload data register value before the timer operation stops. Set the 8-bit timer in this mode when a certain waiting time must be generated.

Note: When setting the count mode, make sure the 8-bit timer counter is stopped.

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

Resets the 8-bit timer.

1 (W): Reset
0 (W): Has no effect
0 (R): Always 0 when read (default)

Writing 1 to this bit presets the reload data in the counter.

D0 PRUN: Timer Run/Stop Control Bit

Controls the timer's Run/Stop state.

1 (R/W): Run
0 (R/W): Stop (default)

The timer starts counting by writing 1 to PRUN and stops counting by writing 0.
In the stop state, the counter data is retained until the timer is reset or placed in a run state.

12.11 Precautions

- Before the 8-bit timer can start counting, the prescaler must be run.
- When setting the count clock or count mode, make sure the 8-bit timer is turned off.

13 PWM & Capture Timer (T16E)

13.1 Outline of the PWM & Capture Timer

The S1C17701 incorporates one channel of PWM & capture timer.
Figure 13.1.1 shows the structure of the PWM & capture timer.

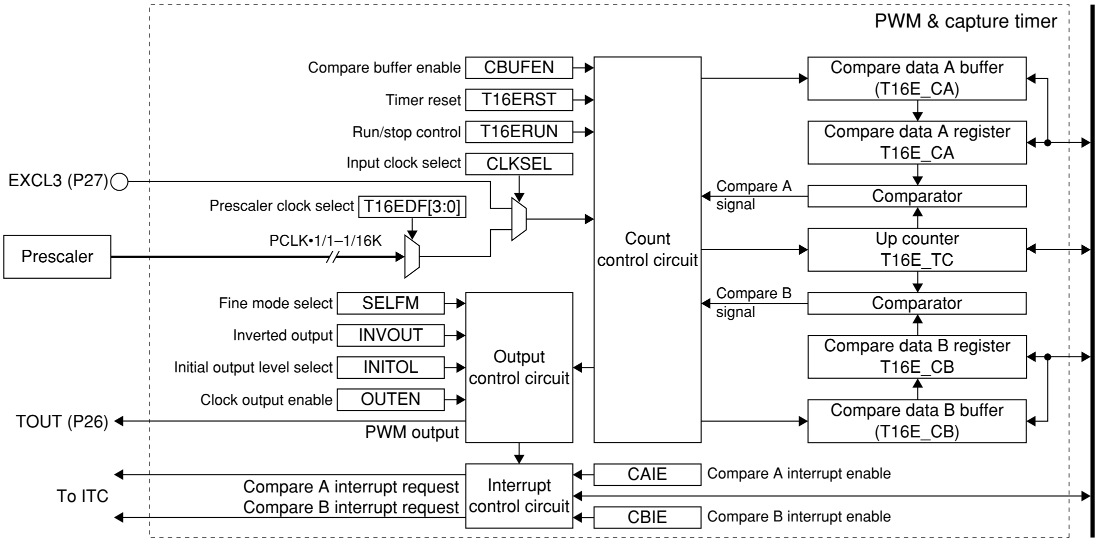


Figure 13.1.1 Structure of PWM & Capture Timer

In the PWM & capture timer, a 16-bit up-counter (T16E_TC register), as well as two 16-bit compare data registers (T16E_CA and T16E_CB registers) and their buffers, are provided.

The 16-bit counter can be reset to 0 or set with an initial value by software and counts up using the prescaler output clock or an external signal input from the P27 port. The counter value can be read by software.

The compare data A and B registers are used to store the data to be compared with the content of the up-counter. This register can be directly read and written. Furthermore, compare data can be set via the compare data buffer. In this case, the set value is loaded to the compare data register when the counter is reset by the compare B match signal or software. The software can select whether compare data is written to the compare data register or the buffer.

When the counter value matches to the content of each compare data register, the comparator outputs a signal that controls the interrupt and the output signal. Thus the registers allow interrupt generating intervals and the timer's output clock frequency and duty ratio to be programmed.

13.2 PWM & Capture Timer Operating Mode

The PWM & capture timer has two operating modes.
1. Internal clock mode (counts the internal clock)
2. External clock mode (functions as an event counter)

Use CLKSEL (D3/T16E_CTL register) to select an operating mode.

∗ **CLKSEL**: Input Clock Select Bit in the PWM Timer Control (T16E_CTL) Register (D3/0x5306)

When CLKSEL is set to 0 (default), the timer enters internal clock mode; when it is set to 1, the timer enters external clock mode.

Internal Clock Mode

In internal clock mode, the timer uses the prescaler output clock as the count clock.
Use the T16EDF[3:0] (D[3:0]/T16E_CLK register) to select the count clock from the 15 clocks, PCLK divided by 1 to PCLK divided by 16K, generated by the prescaler.

∗ **T16EDF[3:0]**: Timer Input Clock Select Bits in the PWM Timer Input Clock Select (T16E_CLK) Register (D[3:0]/0x5308)

Table 13.2.1 Selecting a Prescaler Clock

T16EDF[3:0]	Prescaler output clock	T16EDF[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK•1/128
0xe	PCLK•1/16384	0x6	PCLK•1/64
0xd	PCLK•1/8192	0x5	PCLK•1/32
0xc	PCLK•1/4096	0x4	PCLK•1/16
0xb	PCLK•1/2048	0x3	PCLK•1/8
0xa	PCLK•1/1024	0x2	PCLK•1/4
0x9	PCLK•1/512	0x1	PCLK•1/2
0x8	PCLK•1/256	0x0	PCLK•1/1

(Default: 0x0)

Notes:
- Before the PWM & capture timer can start counting in internal clock mode, the prescaler must be run.
- When setting the count clock, make sure the PWM & capture timer counter is stopped.

For controlling the prescaler, see Chapter 9, “Prescaler (PSC).”

External clock mode

In external clock mode, the timer uses the clock or pulses input from the P27 (EXCL3) port as the count clock. Thus the timer can be used as an event counter. The timer operation is the same as internal clock mode except the count clock source.
To input the EXCL3 clock from the P27 port, write 1 to P27MUX (D7/P2_PMUX register) to switch the P27 pin function in advance.

∗ **P27MUX**: P27 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register

The PWM & capture timer counts up at the rising edge of the input signal.

The PWM & capture timer set in this mode does not use the prescaler. If the prescaler clocks are not used in other peripheral modules, the prescaler can be stopped to reduce current consumption.

13.3 Setting/Resetting the Counter Value

To reset the PWM & capture timer, write 1 to the T16ERST bit (D1/T16E_CTL register). This initializes the counter to 0.

* **T16ERST**: Timer Reset Bit in the PWM Timer Control (T16E_CTL) Register (D1/0x5306)

Normally, reset the counter before starting count-up by writing 1 to this control bit.
After the counter starts counting, it will be reset by the hardware when the counter reaches compare data B.

Furthermore, any data can be set to the counter by writing it to T16ETC[15:0] (D[15:0]/T16E_TC register).

* **T16ETC[15:0]**: Counter Data in the PWM Timer Counter Data (T16E_TC) Register (D[15:0]/0x5304)

13.4 Setting Compare Data

Selecting compare data register/buffer

The PWM & capture timer contains two data comparators that allows the count data to be compared with given values. The compare data A and B registers are used to set these values. These registers can be directly read and written.
Furthermore, compare data can be set via the compare data buffer. In this case, the set value is loaded to the compare data register when the counter is reset by the compare B match signal or software (by writing 1 to T16ERST).
Select whether compare data is written to the compare data register or the buffer using CBUFEN (D5/0x5306).

∗ **CBUFEN**: Comparison Buffer Enable Bit in the PWM Timer Control (T16E_CTL) Register (D5/0x5306)

When 1 is written to CBUFEN, the compare data buffer is selected; when 0 is written, the compare data register is selected.
At initial reset, the compare data register is selected.

Writing compare data

Write compare data A to T16ECA[15:0] (D[15:0]/T16E_CA register) and compare data B to T16ECB[15:0] (D[15:0]/T16E_CB register).

∗ **T16ECA[15:0]**: Compare Data A in the PWM Timer Compare Data A (T16E_CA) Register (D[15:0]/0x5300)
∗ **T16ECB[15:0]**: Compare Data B in the PWM Timer Compare Data B (T16E_CB) Register (D[15:0]/0x5302)

When CBUFEN is set to 0, these registers allow direct reading/writing from/to the compare data register.
When CBUFEN is set to 1, these registers are used to read/write from/to the compare data buffer. The content of the buffer is loaded to the compare data register when the counter is reset.
At initial reset, the compare data registers/buffers are set to 0x0.

The timer compares the compare data register and count data and, when the two values are equal, generates a compare match signal. This compare match signal controls the clock output (TOUT signal) to external devices, in addition to generating an interrupt.

The compare data B is also used to reset the counter.
The counter reset period is calculated by the expression below.

$$\text{Counter reset period} = \frac{\text{CB} + 1}{\text{clk\_in}} \text{ [s]}$$

$$\text{Counter reset cycle} = \frac{\text{clk\_in}}{\text{CB} + 1} \text{ [Hz]}$$

CB: Compare data B (T16E_CB register value)
clk_in: Prescaler output clock frequency

13.5 PWM & Capture Timer Run/Stop Control

Before starting the PWM & capture timer, set up the conditions as shown below.

(1) Select an operating mode (input clock). See Section 13.2.
(2) Set clock output conditions. See Section 13.6.
(3) Set up the interrupt level and enable the PWM & capture timer interrupt if the interrupt is used. See Section 13.7.
(4) Set a value to the counter or reset the counter to 0. See Section 13.3.
(5) Set compare data. See Section 13.4.

The PWM & capture timer provides T16ERUN (D0/T16E_CTL register) to run and stop the counter.

∗ **T16ERUN**: Timer Run/Stop Control Bit in the PWM Timer Control (T16E_CTL) Register (D0/0x5306)

The timer starts counting when 1 is written to T16ERUN. The clock input is disabled and the timer stops counting when 0 is written to T16ERUN. This control does not affect the counter data. Even when the timer has stopped counting, the counter retains its count so that the timer can start counting again from that point.
When both T16ERUN and T16ERST are set to 1 at the same time, the timer starts counting after resetting the counter.

If the count of the counter matches the set value of the compare data A register during count-up, the timer outputs the compare A match signal as a cause of interrupt.
When the counter matches compare data B, the timer outputs the compare B match signal as a cause of interrupt and resets the counter. At the same time, the values set in the compare data buffer are loaded to the compare data register if CBUFEN is set to 1. If the interrupt has been enabled, an interrupt request is sent to the interrupt controller (ITC).
The counter continues counting up regardless of which interrupt has occurred. In the case of a compare B match, the counter starts counting beginning with 0.

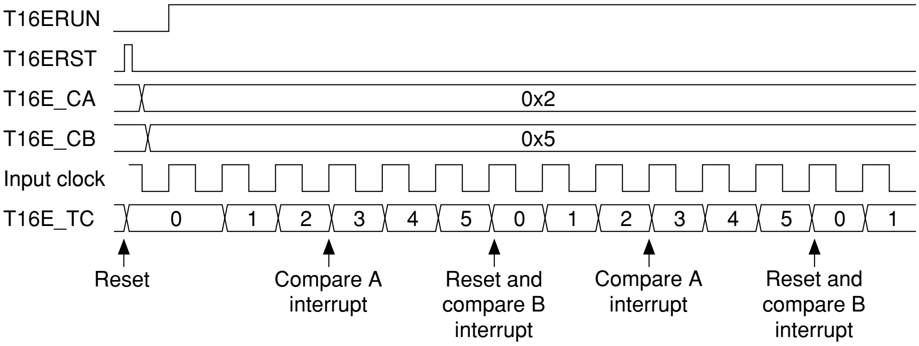


Figure 13.5.1 Basic Operation Timing of Counter

13.6 Controlling Clock Output

The PWM & capture timer can generate the TOUT signal using the compare match signals from the counter. Figure 13.6.1 shows the PWM & capture timer clock output circuit.

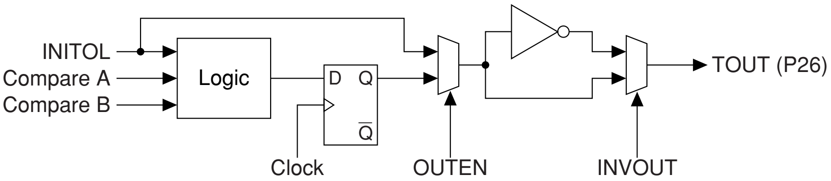


Figure 13.6.1 PWM & Capture Timer Clock Output Circuit

Setting the initial output level

The default output level while the clock output is turned off is 0 (low level). This level can be changed to 1 (high level) using INITOL (D8/T16E_CTL register).

* **INITOL**: Initial Output Level Select Bit in the PWM Timer Control (T16E_CTL) Register (D8/0x5306)

When INITOL is 0 (default), the initial output level is low. When INITOL is set to 1, the initial output level is set to high.
The timer output goes to the initial output level when the timer is reset by writing 1 to T16ERST as well as when the timer output is turned off.

Setting the signal active level

By default, an active high signal (normal low) is generated. This logic can be inverted using INVOUT (D4/T16E_CTL register). When 1 is written to INVOUT, the timer generates an active low (normal high) signal.

* **INVOUT**: Inverse Output Control Bit in the PWM Timer Control (T16E_CTL) Register (D4/0x5306)

Note that the initial output level set by INITOL is inverted when INVOUT is set to 1.
See Figure 13.6.2 for the waveforms.

Setting the output pin

The TOUT signal generated here can be output from the TOUT (P26) pin, enabling a programmable clock or PWM signal to be supplied to external devices.
At initial reset, the P26 pin used for the clock output is set for the I/O port and set in input mode. The pin goes into high-impedance status.
When the pin function is switched to the TOUT output, the pin outputs the level according to the set values of INITOL and INVOUT. The output pin holds this level until the output level changes due to the counter value after the timer output is enabled.

Table 13.6.1 Initial Output Level

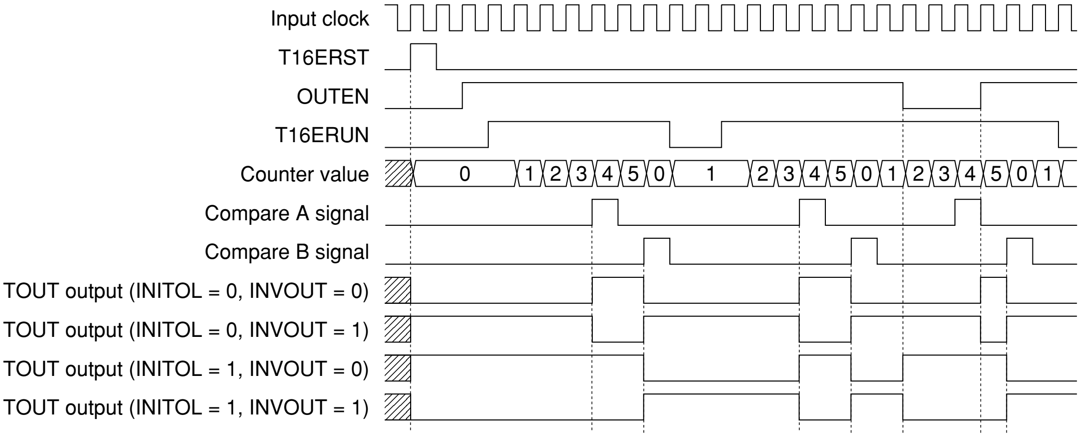
INITOL	INVOUT	Initial output level
1	1	Low
1	0	High
0	1	High
0	0	Low

Starting clock output

To output the TOUT clock, write 1 to OUTEN (D2/T16E_CTL register). Clock output is stopped by writing 0 to OUTEN and goes to the initial output level according to the set values of INITOL and INVOUT.

∗ **OUTEN**: Clock Output Enable Bit in the PWM Timer Control (T16E_CTL) Register (D2/0x5306)

Figure 13.6.2 shows the waveform of the output signal.


Input clock
T16ERST
OUTEN
T16ERUN
Counter value
0
1
2
3
4
5
0
1
2
3
4
5
0
1
2
3
4
5
0
1
Compare A signal
Compare B signal
TOUT output (INITOL = 0, INVOUT = 0)
TOUT output (INITOL = 0, INVOUT = 1)
TOUT output (INITOL = 1, INVOUT = 0)
TOUT output (INITOL = 1, INVOUT = 1)

(When T16E_CA = 3, T16E_CB = 5)

Figure 13.6.2 Waveform of PWM & Capture Timer Output

When INVOUT = 0 (active high)

The timer outputs a low level (initial output level when output is started) until the counter becomes equal to the compare data A set in the T16E_CA register (0x5300). When the counter is incremented to the next value from the compare data A, the output pin goes high and a cause of compare A interrupt occurs. When the counter becomes equal to the compare data B set in the T16E_CB register (0x5302), the counter is reset and the output pin goes low. At the same time a cause of compare B interrupt occurs.

When INVOUT = 1 (active low)

The timer outputs a high level (inverted initial output level when output is started) until the counter becomes equal to the compare data A set in the T16E_CA register (0x5300). When the counter is incremented to the next value from the compare data A, the output pin goes low and a cause of compare A interrupt occurs. When the counter becomes equal to the compare data B set in the T16E_CB register (0x5302), the counter is reset and the output pin goes high. At the same time a cause of compare B interrupt occurs.

Setting fine mode for clock output

By default (after an initial reset), the clock output signal changes at the rising edge of the input clock when compare data A becomes equal to the counter value.

In fine mode, the output signal changes according to bit 0 of compare data A (T16ECA0) when the T16ECA0[15:1] value in the compare data A register becomes equal to the T16ETC[14:0] counter value.
When T16ECA0 is 0, the output signal changes at the rising edge of the input clock.
When T16ECA0 is 1, the output signal changes at the falling edge of the input clock a half cycle from the default setting.

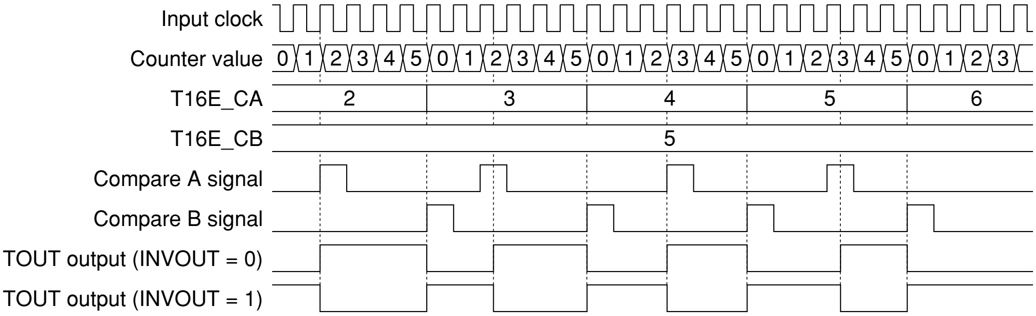


Figure 13.6.3 Clock Output in Fine Mode

As shown in the figure above, in fine mode the output clock duty ratio can be adjusted in the half cycle of the input clock. However, when compare data A is 0, the timer outputs a pulse with a 1-cycle width as the input clock, the same as the default setting.
In fine mode, the maximum value of compare data B is 2^{15} - 1 = 32,767 and the range of compare data A that can be set is 0 to (2 × compare data B - 1).

The fine mode is set using SELFM (D6/T16E_CTL register).

∗ **SELFM**: Fine Mode Select Bit in the PWM Timer Control (T16E_CTL) Register (D6/0x5306)

When 1 is written to SELFM, fine mode is set. At initial reset, the fine mode is disabled.

Precautions

(1) When using the output clock, set compare data as A ≥ 0 and B ≥ 1. The minimum settings are A = 0 and B = 1. In this case, the timer output clock cycle is the input clock × 1/2.

(2) When compare data are set as A > B (or set as A > B × 2 in fine mode), the compare B match signal will be generated but no compare A match signal will be generated. In this case, the timer output signal is fixed at the low (or high when INVOUT = 1).

13.7 PWM & Capture Timer Interrupt

The T16E module can generate the following two types of interrupts:
• Compare A match interrupt
• Compare B match interrupt

The T16E module has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with the two causes of interrupt. To determine the cause of interrupt that has occurred, read the interrupt flags in the T16E module.

Compare A match interrupt

This interrupt request occurs when the count of the counter matches the set value of the compare data A register during count-up, and it sets the interrupt flag CAIF (D0/T16E_IFLG register) in the T16E module to 1.

* **CAIF**: Compare A Interrupt Flag in the PWM Timer Interrupt Flag (T16E_IFLG) Register (D0/0x530c)

Set the CAIE bit (D0/T16E_IMSK register) to 1 when using this interrupt. If CAIE is set to 0 (default), CAIF will not be set to 1 and an interrupt request by this cause will not be sent to the ITC.

* **CAIE**: Compare A Interrupt Enable Bit in the PWM Timer Interrupt Mask (T16E_IMSK) Register (D0/0x530a)

If CAIF is set to 1, the T16E module outputs the interrupt request signal to the ITC. The interrupt request signal sets the PWM & capture timer interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The PWM & capture timer interrupt handler routine should read the CAIF flag to check if the interrupt has occurred due to a compare A match or another cause.
Furthermore, the interrupt handler routine must reset (write 1 to) CAIF in the T16E module, not the PWM & capture timer interrupt flag in the ITC, to clear the cause of interrupt.

Compare B match interrupt

This interrupt request occurs when the count of the counter matches the set value of the compare data B register during count-up, and it sets the interrupt flag CBIF (D1/T16E_IFLG register) in the T16E module to 1.

* **CBIF**: Compare B Interrupt Flag in the PWM Timer Interrupt Flag (T16E_IFLG) Register (D1/0x530c)

Set the CBIE bit (D1/T16E_IMSK register) to 1 when using this interrupt. If CBIE is set to 0 (default), CBIF will not be set to 1 and an interrupt request by this cause will not be sent to the ITC.

* **CBIE**: Compare B Interrupt Enable Bit in the PWM Timer Interrupt Mask (T16E_IMSK) Register (D1/0x530a)

If CBIF is set to 1, the T16E module outputs the interrupt request signal to the ITC. The interrupt request signal sets the PWM & capture timer interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The PWM & capture timer interrupt handler routine should read the CBIF flag to check if the interrupt has occurred due to a compare B match or another cause.
Furthermore, the interrupt handler routine must reset (write 1 to) CBIF in the T16E module, not the PWM & capture timer interrupt flag in the ITC, to clear the cause of interrupt.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the CAIF or CBIF flag before the compare A match or compare B match interrupt is enabled using CAIE or CBIE.

ITC registers for PWM & capture timer interrupt

When a compare match whose interrupt is enabled occurs according to the interrupt condition settings shown above, the T16E module asserts the interrupt signal sent to the ITC.
To generate a PWM & capture timer interrupt, set the interrupt level and enable the interrupt using the ITC registers.
The following shows the control bits for the PWM & capture timer interrupt in the ITC.

Interrupt flag in the ITC

∗ **EIFT7**: PWM & Capture Timer Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D7/0x4300)

Interrupt enable bit in the ITC

∗ **EIEN7**: PWM & Capture Timer Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D7/0x4302)

Interrupt level setup bits in the ITC

∗ **EILV7[2:0]**: T16E Interrupt Level Bits in the External Interrupt Level Setup (ITC_ELV3) Register 3 (D[10:8]/0x430c)

Interrupt trigger mode select bit in the ITC (fixed at 1)

∗ **EITG7**: T16E Interrupt Trigger Mode Select Bit in the External Interrupt Level Setup (ITC_ELV3) Register 3 (D12/0x430c)

When a compare match whose interrupt is enabled in the T16E module occurs, EIFT7 is set to 1. If EIEN7 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the PWM & capture timer interrupt, set EIEN7 to 0. EIFT7 is always set to 1 by the interrupt signal sent from the T16E module, regardless of how EIEN7 is set (even when set to 0).
EILV7[2:0] sets the interrupt level (0 to 7) of the PWM & capture timer interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The PWM & capture timer interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to Chapter 6, "Interrupt Controller (ITC)."

Note: The settings shown below are required to manage the cause-of-interrupt occurrence status using the interrupt flags in the T16E module.

1. Set the PWM & capture timer interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the CAIF or CBIF interrupt flag of the T16E module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

Interrupt vector

The following shows the vector number and vector address for the PWM & capture timer interrupt:

Vector number: 11 (0x0b)
Vector address: 0x802c

13.8 Details of Control Registers

Table 13.8.1 List of PWM & Capture Timer Registers

Address	Register name		Function
0x5300	T16E_CA	PWM Timer Compare Data A Register	Sets compare data A.
0x5302	T16E_CB	PWM Timer Compare Data B Register	Sets compare data B.
0x5304	T16E_TC	PWM Timer Counter Data Register	Counter data
0x5306	T16E_CTL	PWM Timer Control Register	Sets the timer mode and starts/stops the timer.
0x5308	T16E_CLK	PWM Timer Input Clock Select Register	Selects a prescaler output clock.
0x530a	T16E_IMSK	PWM Timer Interrupt Mask Register	Enables/disables interrupt.
0x530c	T16E_IFLG	PWM Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The following describes each PWM & capture timer register. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5300: PWM Timer Compare Data A Register (T16E_CA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Compare Data A Register (T16E_CA)	**0x5300** (16 bits)	D15–0	**T16ECA[15:0]**	Compare data A T16ECA15 = MSB T16ECA0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] T16ECA[15:0]: Compare Data A

Sets the compare data A for the PWM & capture timer. (Default: 0x0)

When CBUFEN (D5/T16E_CTL register) is set to 0, compare data is directly read or writing from/to the compare data A register.

When CBUFEN is set to 1, compare data is read or written from/to the compare data A buffer through this register. The content of the buffer is loaded to the compare data A register when the counter is reset.

The data set in this register is compared with the counter data. When the contents match, a cause of compare A interrupt is generated and the output signal rises (INVOUT (D4/T16E_CTL register) = 0) or falls (INVOUT = 1). This does not affect the counter value and count-up operation.

0x5302: PWM Timer Compare Data B Register (T16E_CB)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Compare Data B Register (T16E_CB)	**0x5302** (16 bits)	D15–0	**T16ECB[15:0]**	Compare data B T16ECB15 = MSB T16ECB0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] T16ECB[15:0]: Compare Data B

Sets the compare data B for the PWM & capture timer. (Default: 0x0)

When CBUFEN (D5/T16E_CTL register) is set to 0, compare data is directly read or writing from/to the compare data B register.

When CBUFEN is set to 1, compare data is read or written from/to the compare data B buffer through this register. The content of the buffer is loaded to the compare data B register when the counter is reset.

The data set in this register is compared with the counter data. When the contents match, a cause of compare B interrupt is generated and the output signal rises (INVOUT (D4/T16E_CTL register) = 0) or falls (INVOUT = 1). Furthermore, the counter is reset to 0.

0x5304: PWM Timer Counter Data Register (T16E_TC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Counter Data Register (T16E_TC)	**0x5304** (16 bits)	D15–0	**T16ETC[15:0]**	Counter data T16ETC15 = MSB T16ETC0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] T16ETC[15:0]: Counter Data

The counter data can be read from this register. (Default: 0x0)
Furthermore, data can be set to the counter by writing it to this register.

0x5306: PWM Timer Control Register (T16E_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
PWM Timer Control Register (T16E_CTL)	**0x5306** (16 bits)	D15–9	–	reserved	–				–	–	0 when being read.
		D8	**INITOL**	Initial output level	1	High	0	Low	0	R/W	
		D7	–	reserved	–				–	–	0 when being read.
		D6	**SELFM**	Fine mode select	1	Fine mode	0	Normal mode	0	R/W	
		D5	**CBUFEN**	Comparison buffer enable	1	Enable	0	Disable	0	R/W	
		D4	**INVOUT**	Inverse output	1	Invert	0	Normal	0	R/W	
		D3	**CLKSEL**	Input clock select	1	External	0	Internal	0	R/W	
		D2	**OUTEN**	Clock output enable	1	Enable	0	Disable	0	R/W	
		D1	**T16ERST**	Timer reset	1	Reset	0	Ignored	0	W	0 when being read.
		D0	**T16ERUN**	Timer run/stop control	1	Run	0	Stop	0	R/W	

D[15:9] Reserved

D8 INITOL: Initial Output Level Bit

Selects an initial output level for timer output.

1 (R/W): High

0 (R/W): Low (default)

The timer output pin goes to the initial output level set using this bit when the timer output is turned off by writing 0 to OUTEN (D2) or when the timer is reset by writing 1 to T16ERST (D1). However, this level is inverted if INVOUT (D4) is set to 1.

D7 Reserved

D6 SELFM: Fine Mode Select Bit

Sets fine mode for clock output.

1 (R/W): Fine mode

0 (R/W): Normal output (default)

When SELFM is set to 1, clock output is set in fine mode which allows adjustment of the output signal duty ratio in units of a half cycle for the input clock.

When SELFM is set to 0, normal clock output will be performed.

D5 CBUFEN: Comparison Buffer Enable Bit

Enables or disables writing to the compare data buffer.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When CBUFEN is set to 1, compare data is read and written from/to the compare data buffer. The content of the buffer is loaded to the compare data register when the counter is reset by the software or the compare B signal.

When CBUFEN is set to 0, compare data is read and written from/to the compare data register.

D4 INVOUT: Inverse Output Control Bit

Selects a logic of the output signal.

1 (R/W): Inverted (active low)

0 (R/W): Normal (active high) (default)

By writing 1 to INVOUT, an active-low signal (off level = high) is generated for the TOUT output. When INVOUT is set to 0, an active-high signal (off level = low) is generated.

Writing 1 to this bit inverts the initial output level set using INITOL (D8) as well.

D3 CLKSEL: Input Clock Select Bit

Selects the input clock for the timer.
1 (R/W): External clock
0 (R/W): Internal clock (default)

The internal clock (prescaler output) is selected for the input clock of the timer by writing 0 to CLKSEL. An external clock (one that is fed from the EXCL3 (P27) pin) is selected by writing 1, and the timer functions as an event counter.

Note: To input the EXCL3 clock from the P27 port, write 1 to P27MUX (D7/P2_PMUX register) to switch the P27 pin function in advance.

∗ **P27MUX**: P27 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register

D2 OUTEN: Clock Output Enable Bit

Controls the output of the TOUT signal (timer output clock).
1 (R/W): Enable
0 (R/W): Disable (default)

The TOUT signal is output from the TOUT (P26) output pin by writing 1 to OUTEN. Clock output is stopped by writing 0 to OUTEN and goes to the off level according to the set values of INVOUT (D4) and INITOL (D8). In this case, the P26 pin must be set for the TOUT output using the P26 port function select register before outputting the TOUT signal.

D1 T16ERST: Timer Reset Bit

Resets the counter.
1 (W): Reset
0 (W): Has no effect
0 (R): Always 0 when read (default)

Writing 1 to T16ERST resets the counter in the PWM & capture timer.

D0 T16ERUN: Timer Run/Stop Control Bit

Controls the timer's Run/Stop state.
1 (R/W): Run
0 (R/W): Stop (default)

The PWM & capture timer starts counting up by writing 1 to T16ERUN and stops by writing 0.
In Stop state, the counter data is retained until the timer is reset or placed in a Run state. By changing states from Stop to Run, the timer can restart counting beginning at the retained count.

0x5308: PWM Timer Input Clock Select Register (T16E_CLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
PWM Timer Input Clock Select Register (T16E_CLK)	**0x5308** (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	**T16EDF[3:0]**	Timer input clock select (Prescaler output clock)	T16EDF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			

D[15:4] Reserved

D[3:0] T16EDF[3:0]: Timer Input Clock Select Bits

These bits select the count clock of the PWM & capture timer from 15 prescaler output clocks.

Table 13.8.2 Selecting the Count Clock

T16EDF[3:0]	Prescaler output clock	T16EDF[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK•1/128
0xe	PCLK•1/16384	0x6	PCLK•1/64
0xd	PCLK•1/8192	0x5	PCLK•1/32
0xc	PCLK•1/4096	0x4	PCLK•1/16
0xb	PCLK•1/2048	0x3	PCLK•1/8
0xa	PCLK•1/1024	0x2	PCLK•1/4
0x9	PCLK•1/512	0x1	PCLK•1/2
0x8	PCLK•1/256	0x0	PCLK•1/1

(Default: 0x0)

Note: When setting the count clock, make sure the PWM & capture timer counter is stopped.

0x530a: PWM Timer Interrupt Mask Register (T16E_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
PWM Timer Interrupt Mask Register (T16E_IMSK)	**0x530a** (16 bits)	D15–2	–	reserved	–				–	–	0 when being read.
		D1	**CBIE**	Compare B interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**CAIE**	Compare A interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:2] Reserved

D1 CBIE: Compare B Interrupt Enable Bit

Enables/disables the compare B interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

Setting CBIE to 1 enables the compare B interrupt; setting to 0 disables the interrupt.
In addition, it is necessary to set the PWM & capture timer interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

D0 CAIE: Compare A Interrupt Enable Bit

Enables/disables the compare A interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

Setting CAIE to 1 enables the compare A interrupt; setting to 0 disables the interrupt.
In addition, it is necessary to set the PWM & capture timer interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

0x530c: PWM Timer Interrupt Flag Register (T16E_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
PWM Timer Interrupt Flag Register (T16E_IFLG)	**0x530c** (16 bits)	D15–2	–	reserved	–				–	–	0 when being read.
		D1	**CBIF**	Compare B interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D0	**CAIF**	Compare A interrupt flag					0	R/W	

D[15:2] Reserved

D1 CBIF: Compare B Interrupt Flag

This is the interrupt flag to indicate the compare B interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

CBIF is the interrupt flag for the compare B interrupt. The interrupt flag is set to 1 when the count of the counter matches the set value of the compare data B register during count-up if CBIE (D1/T16E_IMSK) has been set to 1. At the same time, the PWM & capture timer interrupt request signal is output to the ITC. The interrupt request signal sets the PWM & capture timer interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

D0 CAIF: Compare A Interrupt Flag

This is the interrupt flag to indicate the compare A interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

CAIF is the interrupt flag for the compare A interrupt. The interrupt flag is set to 1 when the count of the counter matches the set value of the compare data A register during count-up if CAIE (D0/T16E_IMSK) has been set to 1. At the same time, the PWM & capture timer interrupt request signal is output to the ITC. The interrupt request signal sets the PWM & capture timer interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

The settings shown below are required to manage the cause-of-interrupt occurrence status using this register.

1. Set the PWM & capture timer interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the CAIF or CBIF interrupt flag of the T16E module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

The CAIF and CBIF flags are reset by writing 1.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the CAIF or CBIF flag before the compare A match or compare B match interrupt is enabled using CAIE (D0/T16E_IMSK) or CBIE (D1/T16E_IMSK).

13.9 Precautions

- Before the PWM & capture timer can start counting, the prescaler must be run.
- When setting the count clock, make sure the PWM & capture timer is turned off.
- When using the output clock, set compare data as A ≥ 0 and B ≥ 1. The minimum settings are A = 0 and B = 1. In this case, the timer output clock cycle is the input clock × 1/2.
- When compare data are set as A > B (or set as A > B × 2 in fine mode), the compare B match signal will be generated but no compare A match signal will be generated. In this case, the timer output signal is fixed at the low (or high when INVOUT = 1).
- To avoid occurrence of unnecessary interrupts, be sure to reset CAIF (D0/T16E_IFLG register) or CBIF (D1/T16E_IFLG register) flag before the compare A match or compare B match interrupt is enabled using CAIE (D0/T16E_IMSK register) or CBIE (D1/T16E_IMSK register).

14 8-bit OSC1 Timer (T8OSC1)

14.1 Outline of the 8-bit OSC1 Timer

The S1C17701 incorporates one channel of 8-bit OSC1 timer that uses OSC1 as its clock source.
Figure 14.1.1 shows the structure of the 8-bit OSC1 timer.

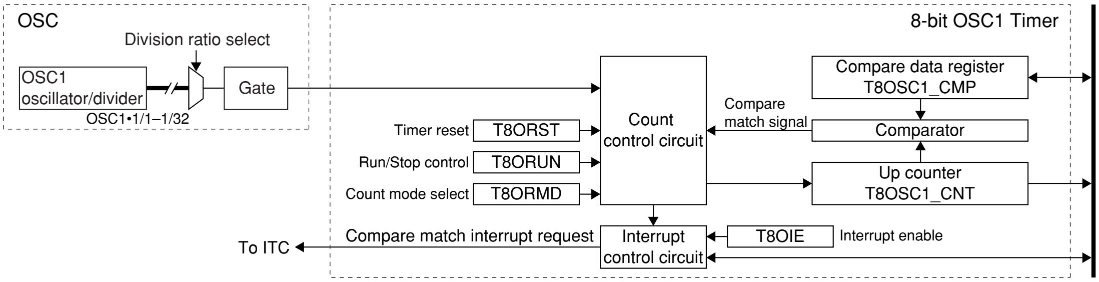


Figure 14.1.1 Structure of 8-bit OSC1 Timer

In the 8-bit OSC1 timer, an 8-bit up-counter (T8OSC1_CNT register) and an 8-bit compare data register (T8OSC1_CMP register) are provided.

The 8-bit counter can be reset to 0 with software and counts up using an divided OSC1 clock (OSC1•1/1–OSC1• 1/32). The counter value can be read by software.

The compare data register is used to store the data to be compared with the content of the up-counter. When the counter value matches to the content of the compare data register, the comparator outputs a signal that controls the interrupt. Thus the register allows interrupt generating interval to be programmed.

14.2 Count Mode of the 8-bit OSC1 Timer

The 8-bit OSC1 timer has two count modes: repeat mode and one-shot mode. It can be selected using the T8ORMD (D1/T8OSC1_CTL register).

∗ **T8ORMD**: Count Mode Select Bit in the 8-bit OSC1 Timer Control (T8OSC1_CTL) Register (D1/0x50c0)

Repeat mode (T8ORMD = 0, default)

The 8-bit OSC1 timer is set in repeat mode when T8ORMD is set to 0.

In this mode, the 8-bit OSC1 timer does not stop after it starts counting until the application program stops the timer. When the counter value matches to the compare data, the timer resets the counter and continues counting. At the same time, the timer outputs the interrupt signal. Set the 8-bit OSC1 timer in this mode when generating periodical interrupts with a given interval.

One-shot mode (T8ORMD = 1)

The 8-bit OSC1 timer is set in one-shot mode when T8ORMD is set to 1.

In this mode, the 8-bit OSC1 timer automatically stops counting when the counter value matches to the compare data, so only one interrupt can be generated after starting the timer. When a compare match occurs, the counter is reset before the timer operation stops. Set the 8-bit OSC1 timer in this mode when a certain waiting time must be generated.

Note: When setting the count mode, make sure the 8-bit OSC1 timer counter is stopped.

14.3 Count Clock

The 8-bit OSC1 timer uses a divided OSC1 clock output from the OSC module as the count clock. The OSC module divides the OSC1 clock by 1 to 32 to generate 6 clocks. Select one of these clocks using the T8O1CK[2:0] bits (D[3:1]/OSC_T8OSC1 register).

∗ **T8O1CK[2:0]**: T8OSC1 Clock Division Ratio Select Bits in the T8OSC1 Clock Control (OSC_T8OSC1) Register (D[3:1]/0x5065)

Table 14.3.1 Selecting the Count Clock

T8O1CK[2:0]	Division ratio
0x7–0x6	Reserved
0x5	OSC1•1/32
0x4	OSC1•1/16
0x3	OSC1•1/8
0x2	OSC1•1/4
0x1	OSC1•1/2
0x0	OSC1•1/1

(Default: 0x0)

Use the T8O1CE bit (D0/OSC_T8OSC1 register) to control the clock supply to the 8-bit OSC1 timer. T8O1CE is set to 0 by default, so the clock is disabled for supplying. When the T8O1CE is set to 1, the clock selected as above is supplied to the 8-bit OSC1 timer. If the 8-bit OSC1 timer does not need to operate, stop the clock supply to reduce the current consumption.

∗ **T8O1CE**: T8OSC1 Clock Enable Bit in the T8OSC1 Clock Control (OSC_T8OSC1) Register (D0/0x5065)

Note: When setting the count clock, make sure the 8-bit OSC1 timer counter is stopped.

For control of the clock, see Chapter 7, "Oscillator (OSC)."

14.4 Resetting the 8-bit OSC1 Timer

To reset the 8-bit OSC1 timer, write 1 to the T8ORST bit (D4/T8OSC1_CTL register). This initializes the counter to 0.

∗ **T8ORST**: Timer Reset Bit in the 8-bit OSC1 Timer Control (T8OSC1_CTL) Register (D4/0x50c0)

Normally, reset the counter before starting count-up by writing 1 to this control bit.
After the counter starts counting, it will be reset by the hardware when the counter reaches compare data.

14.5 Setting Compare Data

Write compare data to T8OCMP[7:0] (D[7:0]/T8OSC1_CMP register).

∗ **T8OCMP[7:0]**: Compare Data Bits in the 8-bit OSC1 Timer Compare Data (T8OSC1_CMP) Register (D[7:0]/0x50c2)

At initial reset, the compare data register is set to 0x0.

The timer compares the compare data register and count data and, when the two values are equal, resets the counter and generates a compare match signal. This compare match signal is used to generate an interrupt.
The compare match period is calculated by the expression below.

$$\text{Compare match period} = \frac{\text{CMP} + 1}{\text{clk\_in}} \text{ [s]}$$

$$\text{Compare match cycle} = \frac{\text{clk\_in}}{\text{CMP} + 1} \text{ [Hz]}$$

CMP: Compare data (T8OSC1_CMP register value)
clk_in: 8-bit OSC1 timer count clock frequency

14.6 8-bit OSC1 Timer Run/Stop Control

Before starting the 8-bit OSC1 timer, set up the conditions as shown below.
(1) Select a count mode (one-shot or repeat). See Section 14.2.
(2) Select the operating clock. See Section 14.3.
(3) Set up the interrupt level and enable the 8-bit OSC1 timer interrupt if the interrupt is used. See Section 14.7.
(4) Reset the timer. See Section 14.4.
(5) Set compare data. See Section 14.5.

The 8-bit OSC1 timer provides T8ORUN (D0/T8OSC1_CTL register) to run and stop the counter.

* **T8ORUN**: Timer Run/Stop Control Bit in the 8-bit OSC1 Timer Control (T8OSC1_CTL) Register (D0/0x50c0)

The timer starts counting when 1 is written to T8ORUN. The clock input is disabled and the timer stops counting when 0 is written to T8ORUN. This control does not affect the counter data. Even when the timer has stopped counting, the counter retains its count so that the timer can start counting again from that point.
When both T8ORUN and T8ORST are set to 1 at the same time, the timer starts counting after resetting the counter.

If the count of the counter matches the set value of the compare data register during count-up, the timer outputs the compare match signal as a cause of interrupt. At the same time, the counter is reset to 0. If the interrupt has been enabled, an interrupt request is sent to the interrupt controller (ITC).
If the timer is set in one-shot mode, the timer stops counting.
If the timer is set in repeat mode, the timer continues counting from the counter value 0.

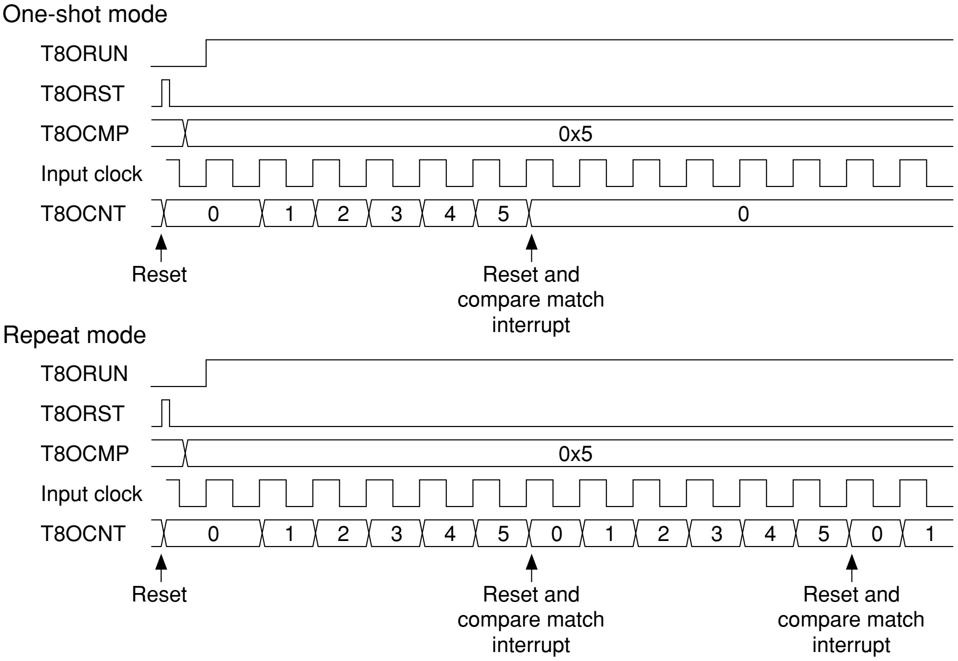


Figure 14.6.1 Basic Operation Timing of Counter

14.7 8-bit OSC1 Timer Interrupt

The T8OSC1 module is able to output an interrupt request signal to the interrupt controller (ITC) when a compare match occurs.

Compare match interrupt

This interrupt request occurs when the count of the counter matches the set value of the compare data register during count-up, and it sets the interrupt flag T8OIF (D0/T8OSC1_IFLG register) in the T8OSC1 module to 1.

∗ **T8OIF**: 8-bit OSC1 Timer Interrupt Flag in the 8-bit OSC1 Timer Interrupt Flag (T8OSC1_IFLG) Register (D0/0x50c4)

Set the T8OIE (D0/T8OSC1_IMSK register) to 1 when using this interrupt. If T8OIE is set to 0 (default), T8OIF will not be set to 1 and an interrupt request by this cause will not be sent to the ITC.

∗ **T8OIE**: 8-bit OSC1 Timer Interrupt Enable Bit in the 8-bit OSC1 Timer Interrupt Mask (T8OSC1_IMSK) Register (D0/0x50c3)

If T8OIF is set to 1, the T8OSC1 module outputs the interrupt request signal to the ITC. The interrupt request signal sets the 8-bit OSC1 timer interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The interrupt handler routine must reset (write 1 to) T8OIF in the T8OSC1 module, not the 8-bit OSC1 timer interrupt flag in the ITC, to clear the cause of interrupt.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the T8OIF flag before the compare match interrupt is enabled using T8OIE.

ITC registers for 8-bit OSC1 timer interrupt

When a compare match occurs according to the interrupt condition settings shown above, the 8-bit OSC1 timer asserts the interrupt signal sent to the ITC.
To generate an 8-bit OSC1 timer interrupt, set the interrupt level and enable the interrupt using the ITC registers.
The following shows the control bits for the 8-bit OSC1 timer interrupt in the ITC.

Interrupt flag in the ITC

∗ **EIFT4**: 8-bit OSC1 Timer Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D4/0x4300)

Interrupt enable bit in the ITC

∗ **EIEN4**: 8-bit OSC1 Timer Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D4/0x4302)

Interrupt level setup bits in the ITC

∗ **EILV4[2:0]**: T8OSC1 Interrupt Level Bits in the External Interrupt Level Setup (ITC_ELV2) Register 2 (D[2:0]/0x430a)

Interrupt trigger mode select bit in the ITC (fixed at 1)

∗ **EITG4**: T8OSC1 Interrupt Trigger Mode Select Bit in the External Interrupt Level Setup (ITC_ELV2) Register 2 (D4/0x430a)

When a compare match whose interrupt is enabled in the T8OSC1 module occurs, EIFT4 is set to 1. If EIEN4 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the 8-bit OSC1 timer interrupt, set EIEN4 to 0. EIFT4 is always set to 1 by the interrupt signal sent from the T8OSC1 module, regardless of how EIEN4 is set (even when set to 0).
EILV4[2:0] sets the interrupt level (0 to 7) of the 8-bit OSC1 timer interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The 8-bit OSC1 timer interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Chapter 6, "Interrupt Controller (ITC)."

Note: The settings shown below are required to manage the cause-of-interrupt occurrence status using the interrupt flag in the T8OSC1 module.

1. Set the 8-bit OSC1 timer interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the T8OIF interrupt flag of the T8OSC1 module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

Interrupt vector

The following shows the vector number and vector address for the 8-bit OSC1 timer interrupt:

Vector number: 8 (0x08)
Vector address: 0x8020

14.8 Details of Control Registers

Table 14.8.1 List of 8-bit OSC1 Timer Registers

Address	Register name		Function
0x50c0	T8OSC1_CTL	8-bit OSC1 Timer Control Register	Sets the timer mode and starts/stops the timer.
0x50c1	T8OSC1_CNT	8-bit OSC1 Timer Counter Data Register	Counter data
0x50c2	T8OSC1_CMP	8-bit OSC1 Timer Compare Data Register	Sets compare data.
0x50c3	T8OSC1_IMSK	8-bit OSC1 Timer Interrupt Mask Register	Enables/disables interrupt.
0x50c4	T8OSC1_IFLG	8-bit OSC1 Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The following describes each 8-bit OSC1 timer register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x50c0: 8-bit OSC1 Timer Control Register (T8OSC1_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1 Timer Control Register (T8OSC1_CTL)	0x50c0 (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	T8ORST	Timer reset	1	Reset	0	Ignored	0	W	
		D3–2	–	reserved	–				–	–	
		D1	T8ORMD	Count mode select	1	One shot	0	Repeat	0	R/W	
		D0	T8ORUN	Timer run/stop control	1	Run	0	Stop	0	R/W	

D[7:5] Reserved

D4 T8ORST: Timer Reset Bit

Resets the 8-bit OSC1 timer.

1 (W): Reset
0 (W): Has no effect
0 (R): Always 0 when read (default)

Writing 1 to this bit resets the counter to 0.

D[3:2] Reserved

D1 T8ORMD: Count Mode Select Bit

Selects the count mode of the 8-bit OSC1 timer.

1 (R/W): One-shot mode
0 (R/W): Repeat mode (default)

The 8-bit OSC1 timer is set in repeat mode when T8ORMD is set to 0. In this mode, the 8-bit OSC1 timer does not stop after it starts counting until the application program stops the timer. When the counter value matches to the compare data, the timer resets the counter and continues counting. At the same time, the timer outputs the interrupt signal. Set the 8-bit OSC1 timer in this mode when generating periodical interrupts with a given interval.

The 8-bit OSC1 timer is set in one-shot mode when T8ORMD is set to 1. In this mode, the 8-bit OSC1 timer automatically stops counting when the counter value matches to the compare data, so only one interrupt can be generated after starting the timer. When a compare match occurs, the counter is reset before the timer operation stops. Set the 8-bit OSC1 timer in this mode when a certain waiting time must be generated.

Note: When setting the count mode, make sure the 8-bit OSC1 timer counter is stopped.

D0 T8ORUN: Timer Run/Stop Control Bit

Controls the timer's Run/Stop state.

1 (R/W): Run
0 (R/W): Stop (default)

The timer starts counting by writing 1 to T8ORUN and stops counting by writing 0.
In the stop state, the counter data is retained until the timer is reset or placed in a run state.

0x50c1: 8-bit OSC1 Timer Counter Data Register (T8OSC1_CNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer Counter Data Register (T8OSC1_CNT)	**0x50c1** (8 bits)	D7–0	**T8OCNT[7:0]**	Timer counter data T8OCNT7 = MSB T8OCNT0 = LSB	0x0 to 0xff	0x0	R	

D[7:0] T8OCNT[7:0]: Counter Data

The counter data can be read from this register. (Default: 0x0)

This is a read-only register, so the writing operation is invalid.

Note: If this register is read while the counter is running, the read value may not represent the current counter value (an indefinite value may be read out).

The counter value should be obtained by one of the following procedures:

- Read the counter value after stopping the counter.
- Read the counter value twice to determine that both read results are the same and that the read value is significant.

0x50c2: 8-bit OSC1 Timer Compare Data Register (T8OSC1_CMP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer Compare Data Register (T8OSC1_CMP)	**0x50c2** (8 bits)	D7–0	**T8OCMP[7:0]**	Compare data T8OCMP7 = MSB T8OCMP0 = LSB	0x0 to 0xff	0x0	R/W	

D[7:0] T8OCMP[7:0]: Compare Data

Sets the compare data for the 8-bit OSC1 timer. (Default: 0x0)

The data set in this register is compared with the counter data. When the contents match, a cause of compare interrupt is generated. At the same time, the counter is reset to 0.

0x50c3: 8-bit OSC1 Timer Interrupt Mask Register (T8OSC1_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1 Timer Interrupt Mask Register (T8OSC1_IMSK)	**0x50c3** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**T8OIE**	8-bit OSC1 timer interrupt enable	1	Enable	0	Disable	0	R/W	

D[7:1] Reserved

D0 T8OIE: 8-bit OSC1 Timer Interrupt Enable Bit

Enables/disables the compare match interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

Setting T8OIE to 1 enables the 8-bit OSC1 timer to request interrupts to the ITC; setting to 0 disables the interrupt.

In addition, it is necessary to set the 8-bit OSC1 timer interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

0x50c4: 8-bit OSC1 Timer Interrupt Flag Register (T8OSC1_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1 Timer Interrupt Flag Register (T8OSC1_IFLG)	**0x50c4** (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	**T8OIF**	8-bit OSC1 timer interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

D[7:1] Reserved

D0 T8OIF: 8-bit OSC1 Timer Interrupt Flag

This is the interrupt flag to indicate the compare match interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

T8OIF is the interrupt flag for the T8OSC1 module. The interrupt flag is set to 1 when the count of the counter matches the set value of the compare data register during count-up if T8OIE (D0/T8OSC1_IMSK register) has been set to 1. At the same time, the 8-bit OSC1 timer interrupt request signal is output to the ITC. The interrupt request signal sets the 8-bit OSC1 timer interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

The settings shown below are required to manage the cause-of-interrupt occurrence status using this register.

1. Set the 8-bit OSC1 timer interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the T8OIF interrupt flag of the T8OSC1 module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

The T8OIF flag is reset by writing 1.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the T8OIF flag before the compare match interrupt is enabled using T8OIE (D0/T8OSC1_IMSK register).

14.9 Precautions

- Before the 8-bit OSC1 timer can start counting, the 8-bit OSC1 timer clock must be supplied from the OSC module.
- When setting the count clock or count mode, make sure the 8-bit OSC1 timer is turned off.
- To avoid occurrence of unnecessary interrupts, be sure to reset T8OIF (D0/T8OSC1_IFLG register) before the compare match interrupt is enabled using T8OIE (D0/T8OSC1_IMSK register).
- If the counter data register is read while the counter is running, the read value may not represent the current counter value (an indefinite value may be read out).
  To obtain the counter value, read the counter data register after stopping the counter. Or read the counter value twice to determine that both read results are the same and that the read value is significant.

THIS PAGE IS BLANK.

15 Clock Timer (CT)

15.1 Outline of the Clock Timer

The S1C17701 incorporates one channel of clock timer that uses OSC1 as its clock source.

The clock timer contains an 8-bit binary counter that operates with a divided OSC1 clock (256 Hz signal). The counter data bits (128 to 1 Hz) can be read with software.

Furthermore, the clock timer can generate an interrupt using the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals.

Normally, the clock timer is used for various timing functions such as a clock.

Figure 15.1.1 shows the structure of the clock timer.

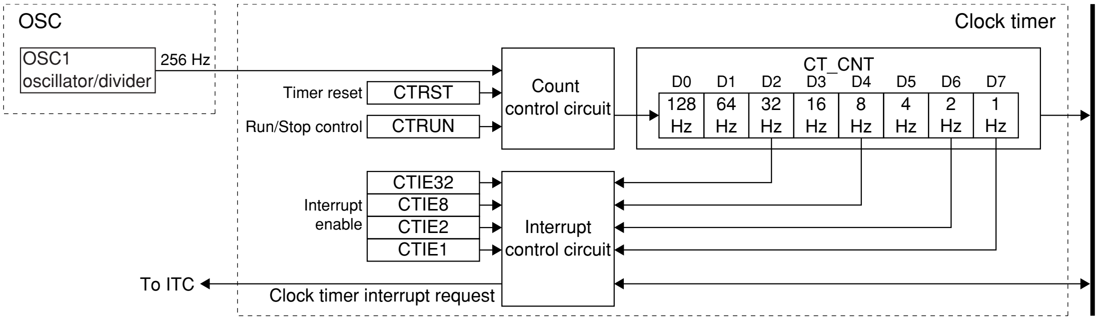


Figure 15.1.1 Structure of Clock Timer

15.2 Operating Clock

The clock timer uses the 256 Hz clock output from the OSC module as its operating clock.

The OSC module generates this clock by dividing the OSC1 clock by 128. Therefore, the clock frequency is 256 Hz when the OSC1 clock frequency is 32.768 kHz. Be aware that the frequencies described in this chapter change if the OSC1 clock has another frequency.

The OSC module does not provide a control bit for the 256 Hz clock. The 256 Hz clock is always supplied to the clock timer when the OSC1 oscillator is on.

For control of the OSC1 oscillator, see Chapter 7, “Oscillator (OSC).”

15.3 Resetting the Clock Timer

To reset the clock timer, write 1 to the CTRST bit (D4/CT_CTL register). This initializes the counter to 0.

∗ **CTRST**: Clock Timer Reset Bit in the Clock Timer Control (CT_CTL) Register (D4/0x5000)

The counter is also initialized to 0 at initial reset.

15.4 Clock Timer Run/Stop Control

Before starting the clock timer, set up the conditions as shown below.
(1) Set up the interrupt level and enable the clock timer interrupt if the interrupt is used. See Section 15.5.
(2) Reset the timer. See Section 15.3.

The clock timer provides CTRUN (D0/CT_CTL register) to run and stop the counter.

∗ **CTRUN**: Clock Timer Run/Stop Control Bit in the Clock Timer Control (CT_CTL) Register (D0/0x5000)

The clock timer starts counting when 1 is written to CTRUN. The clock input is disabled and the timer stops counting when 0 is written to CTRUN. This control does not affect the counter data (CT_CNT register). Even when the timer has stopped counting, the counter retains its count so that the timer can start counting again from that point.
When both CTRUN and CTRST are set to 1 at the same time, the timer starts counting after resetting the counter.

While the timer is running, a cause of interrupt occurs at the falling edge of the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals. If the interrupt has been enabled, an interrupt request is sent to the interrupt controller (ITC).

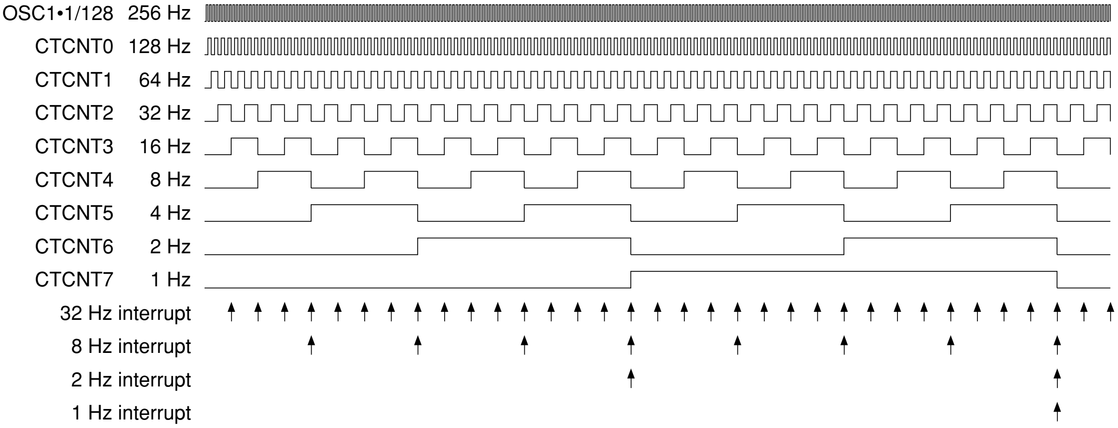


Figure 15.4.1 Timing Chart of Clock Timer

Note: The clock timer is actually made to Run/Stop in sync with the falling edge of the 256 Hz signal after writing to CTRUN. Therefore, when 0 is written to the CTRUN, the timer stops after the counter is incremented. The CTRUN maintains 1 for reading until the timer actually enters Stop status.
Figure 15.4.2 shows the timer operation at start/stop.

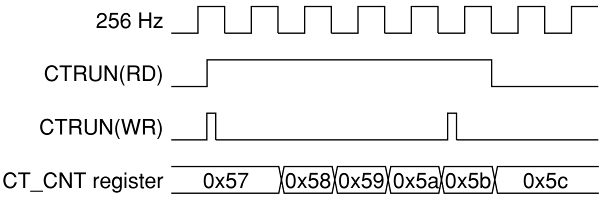


Figure 15.4.2 Clock Timer Start/Stop Operation

15.5 Clock Timer Interrupt

The CT module can generate the following four types of interrupts:

- 32 Hz interrupt
- 8 Hz interrupt
- 2 Hz interrupt
- 1 Hz interrupt

The CT module has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with the four causes of interrupt. To determine the cause of interrupt that has occurred, read the interrupt flags in the CT module.

32 Hz, 8 Hz, 2 Hz, and 1 Hz interrupts

An interrupt request occurs at the falling edge of the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals, and it sets the interrupt flag in the CT module to 1.

∗ **CTIF32**: 32 Hz Interrupt Flag in the Clock Timer Interrupt Flag (CT_IFLG) Register (D3/0x5003)
∗ **CTIF8**: 8 Hz Interrupt Flag in the Clock Timer Interrupt Flag (CT_IFLG) Register (D2/0x5003)
∗ **CTIF2**: 2 Hz Interrupt Flag in the Clock Timer Interrupt Flag (CT_IFLG) Register (D1/0x5003)
∗ **CTIF1**: 1 Hz Interrupt Flag in the Clock Timer Interrupt Flag (CT_IFLG) Register (D0/0x5003)

Set the interrupt enable bit corresponding to the interrupt flag to 1 when using the interrupt. If the interrupt enable bit is set to 0 (default), the interrupt flag will not be set to 1 and an interrupt request by the corresponding signal will not be sent to the ITC.

∗ **CTIE32**: 32 Hz Interrupt Enable Bit in the Clock Timer Interrupt Mask (CT_IMSK) Register (D3/0x5002)
∗ **CTIE8**: 8 Hz Interrupt Enable Bit in the Clock Timer Interrupt Mask (CT_IMSK) Register (D2/0x5002)
∗ **CTIE2**: 2 Hz Interrupt Enable Bit in the Clock Timer Interrupt Mask (CT_IMSK) Register (D1/0x5002)
∗ **CTIE1**: 1 Hz Interrupt Enable Bit in the Clock Timer Interrupt Mask (CT_IMSK) Register (D0/0x5002)

If CTIF∗ is set to 1, the CT module outputs the interrupt request signal to the ITC. The interrupt request signal sets the clock timer interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

The clock timer interrupt handler routine should read the CTIF* flags to check the signal that causes occurrence of the interrupt.

Furthermore, the interrupt handler routine must reset (write 1 to) CTIF∗ in the CT module, not the clock timer interrupt flag in the ITC, to clear the cause of interrupt.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the CTIF∗ flags before the clock timer interrupt is enabled using CTIE∗.

ITC registers for clock timer interrupt

The clock timer asserts the interrupt signal sent to the ITC at the falling edge of the signal whose interrupt is enabled according to the interrupt condition settings shown above. To generate a clock timer interrupt, set the interrupt level and enable the interrupt using the ITC registers.

The following shows the control bits for the clock timer interrupt in the ITC.

Interrupt flag in the ITC

∗ **EIFT3**: Clock Timer Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D3/0x4300)

Interrupt enable bit in the ITC

∗ **EIEN3**: Clock Timer Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D3/0x4302)

Interrupt level setup bits in the ITC

∗ **EILV3[2:0]**: CT Interrupt Level Bits in the External Interrupt Level Setup (ITC_ELV1) Register 1 (D[10:8]/0x4308)

Interrupt trigger mode select bit in the ITC (fixed at 1)

∗ **EITG3**: CT Interrupt Trigger Mode Select Bit in the External Interrupt Level Setup (ITC_ELV1) Register 1 (D12/0x4308)

EIFT3 is set to 1 at the falling edge of the 32/8/2/1 Hz signal whose interrupt is enabled. If EIEN3 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the clock timer interrupt, set EIEN3 to 0. EIFT3 is always set to 1 by the interrupt signal sent from the CT module, regardless of how EIEN3 is set (even when set to 0).
EILV3[2:0] sets the interrupt level (0 to 7) of the clock timer interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The clock timer interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Chapter 6, "Interrupt Controller (ITC)."

Note: The settings shown below are required to manage the cause-of-interrupt occurrence status using the interrupt flags in the CT module.

1. Set the clock timer interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the CTIF∗ interrupt flag of the CT module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

Interrupt vector

The following shows the vector number and vector address for the clock timer interrupt:

Vector number: 7 (0x07)
Vector address: 0x801c

15.6 Details of Control Registers

Table 15.6.1 List of Clock Timer Registers

Address	Register name		Function
0x5000	CT_CTL	Clock Timer Control Register	Resets and starts/stops the timer.
0x5001	CT_CNT	Clock Timer Counter Register	Counter data
0x5002	CT_IMSK	Clock Timer Interrupt Mask Register	Enables/disables interrupt.
0x5003	CT_IFLG	Clock Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The following describes each clock timer register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5000: Clock Timer Control Register (CT_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Clock Timer Control Register (CT_CTL)	**0x5000** (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**CTRST**	Clock timer reset	1	Reset	0	Ignored	0	W	
		D3–1	**–**	reserved	–				–	–	
		D0	**CTRUN**	Clock timer run/stop control	1	Run	0	Stop	0	R/W	

D[7:5] Reserved

D4 CTRST: Clock Timer Reset Bit

Resets the clock timer.

1 (W): Reset

0 (W): Has no effect

0 (R): Always 0 when read (default)

Writing 1 to this bit resets the counter to 0x0.

When the clock timer is reset in Run state, it restarts immediately after resetting. In Stop state, the counter maintains 0x0.

D[3:1] Reserved

D0 CTRUN: Clock Timer Run/Stop Control Bit

Starts/stops the clock timer.

1 (R/W): Run

0 (R/W): Stop (default)

The clock timer starts counting by writing 1 to CTRUN and stops by writing 0.

In Stop state, the counter data is retained until the timer is reset or placed in a Run state.

0x5001: Clock Timer Counter Register (CT_CNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Counter Register (CT_CNT)	**0x5001** (8 bits)	D7–0	**CTCNT[7:0]**	Clock timer counter value	0x0 to 0xff	0	R	

D[7:0] CTCNT[7:0]: Clock Timer Counter Value

The counter data can be read from this register. (Default: 0xff)

This is a read-only register, so the writing operation is invalid.

Each bit corresponds to the frequency as follows:

D7: 1 Hz
D6: 2 Hz
D5: 4 Hz
D4: 8 Hz
D3: 16 Hz
D2: 32 Hz
D1: 64 Hz
D0: 128 Hz

Note: If this register is read while the counter is running, the read value may not represent the current counter value (an indefinite value may be read out).

The counter value should be obtained by one of the following procedures:

- Read the counter value after stopping the counter.
- Read the counter value twice to determine that both read results are the same and that the read value is significant.

0x5002: Clock Timer Interrupt Mask Register (CT_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Clock Timer Interrupt Mask Register (CT_IMSK)	**0x5002** (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**CTIE32**	32 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D2	**CTIE8**	8 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**CTIE2**	2 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**CTIE1**	1 Hz interrupt enable	1	Enable	0	Disable	0	R/W	

This register enables or disables the interrupt requests by the clock timer 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals, individually. Setting CTIE* bit to 1 enables the clock timer interrupt request by the falling edge of the corresponding signal; setting it to 0 disables the interrupt.

In addition, it is necessary to set the clock timer interrupt enable bit in the ITC to interrupt enabled to actually generate an interrupt.

D[7:4] Reserved

D3 CTIE32: 32 Hz Interrupt Enable Bit

Enables/disables the 32 Hz interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

D2 CTIE8: 8 Hz Interrupt Enable Bit

Enables/disables the 8 Hz interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

D1 CTIE2: 2 Hz Interrupt Enable Bit

Enables/disables the 2 Hz interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

D0 CTIE1: 1 Hz Interrupt Enable Bit

Enables/disables the 1 Hz interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

0x5003: Clock Timer Interrupt Flag Register (CT_IFLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Interrupt Flag Register (CT_IFLG)	**0x5003** (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	**CTIF32**	32 Hz interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D2	**CTIF8**	8 Hz interrupt flag		0	R/W	
		D1	**CTIF2**	2 Hz interrupt flag		0	R/W	
		D0	**CTIF1**	1 Hz interrupt flag		0	R/W	

This register indicates the interrupt cause occurrence status by the clock timer 32Hz, 8 Hz, 2 Hz, and 1 Hz signals. When a clock timer interrupt occurs, read the interrupt flag in this register to determine the cause of interrupt that has occurred (or frequency). The CTIF∗ bits are the interrupt flags that correspond to 32 Hz, 8 Hz, 2 Hz, and 1 Hz interrupts respectively. The CTIF∗ bit is set to 1 at the falling edge of the signal if the corresponding CTIE∗ bit (CT_IMSK register) has been set to 1. At the same time, the clock timer interrupt request signal is output to the ITC. The interrupt request signal sets the clock timer interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

The settings shown below are required to manage the cause-of-interrupt occurrence status using this register.

1. Set the clock timer interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the interrupt flag of the CT module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

The CTIF∗ flags are reset by writing 1.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the CTIF∗ flags before the clock timer interrupt is enabled using CTIE∗.

D[7:4] Reserved

D3 CTIF32: 32 Hz Interrupt Flag

This is the interrupt flag to indicate the 32 Hz interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

CTIF32 is set to 1 at the falling edge of the 32 Hz signal only when CTIE32 (D3/CT_IMSK register) has been set to 1.

D2 CTIF8: 8 Hz Interrupt Flag

This is the interrupt flag to indicate the 8 Hz interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

CTIF8 is set to 1 at the falling edge of the 8 Hz signal only when CTIE8 (D2/CT_IMSK register) has been set to 1.

D1 CTIF2: 2 Hz Interrupt Flag

This is the interrupt flag to indicate the 2 Hz interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

CTIF2 is set to 1 at the falling edge of the 2 Hz signal only when CTIE2 (D1/CT_IMSK register) has been set to 1.

D0 CTIF1: 1 Hz Interrupt Flag

This is the interrupt flag to indicate the 1 Hz interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

CTIF1 is set to 1 at the falling edge of the 1 Hz signal only when CTIE1 (D0/CT_IMSK register) has been set to 1.

15.7 Precautions

- Before the clock timer can start counting, the OSC1 oscillator must be turned on.
- To avoid occurrence of unnecessary interrupts, be sure to reset the interrupt flags in the CT_IFLG register before the clock timer interrupt is enabled using the CT_IMSK register.
- The clock timer is actually made to Run/Stop in sync with the falling edge of the 256 Hz signal after writing to CTRUN (D0/CT_CTL register). Therefore, when 0 is written to the CTRUN, the timer stops after the counter is incremented. The CTRUN maintains 1 for reading until the timer actually enters Stop status.
  Figure 15.7.1 shows the timer operation at start/stop.

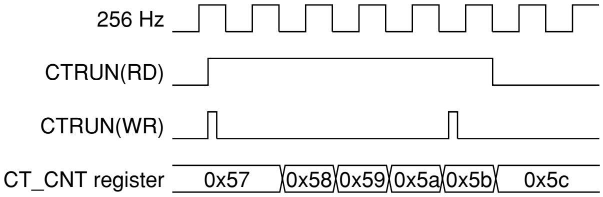


Figure 15.7.1 Clock Timer Start/Stop Operation

- If the `slp` instruction is executed while the clock timer is running (CTRUN = 1), the clock timer will be unstable immediately after SLEEP status is canceled. Therefore, the clock timer should be stopped (CTRUN = 0) before the `slp` instruction is executed to enter SLEEP mode.
- If the counter register is read while the counter is running, the read value may not represent the current counter value (an indefinite value may be read out).
  To obtain the counter value, read the counter register after stopping the counter. Or read the counter value twice to determine that both read results are the same and that the read value is significant.

16 Stopwatch Timer (SWT)

16.1 Outline of the Stopwatch Timer

The S1C17701 incorporates a stopwatch timer that counts 1/100 and 1/10 seconds. The stopwatch timer consists of two 4-bit BCD counters (for 1/100 and 1/10 seconds) that operate with a divided OSC1 clock (256 Hz signal). The counter data can be read with software.

Furthermore, the stopwatch timer can generate an interrupt using the 100 Hz (approximate 100 Hz), 10 Hz (approximate 10 Hz), and 1 Hz signals.

Figure 16.1.1 shows the structure of the stopwatch timer.

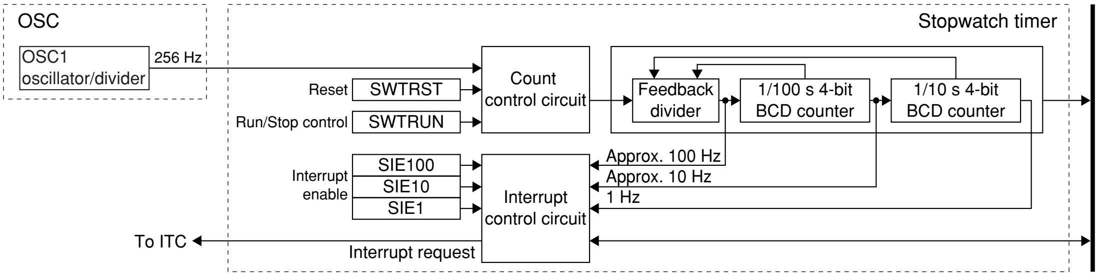


Figure 16.1.1 Structure of Stopwatch Timer

16.2 BCD Counters

The stopwatch timer consists of two 4-bit BCD counters for counting 1/100 seconds and 1/10 seconds.
The count values can be read from the SWT_BCNT register.

1/100-second counter

∗ **BCD100[3:0]**: 1/100 Sec. BCD Counter Value in the Stopwatch Timer BCD Counter (SWT_BCNT) Register (D[3:0]/0x5021)

1/10-second counter

∗ **BCD10[3:0]**: 1/10 Sec. BCD Counter Value in the Stopwatch Timer BCD Counter (SWT_BCNT) Register (D[7:4]/0x5021)

Count-up pattern

To generate 100 Hz, 10 Hz, and 1 Hz signals from the 256 Hz source clock, the stopwatch timer changes the count-up pattern for the counter as shown in Figure 16.2.1 using the feedback divider.

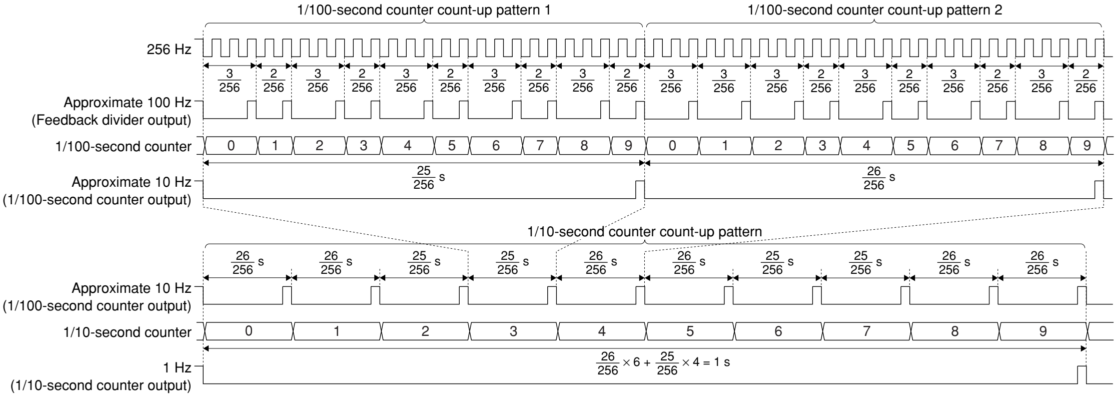

1/100-second counter count-up pattern 1
1/100-second counter count-up pattern 2
256 Hz
3/256
2/256
Approximate 100 Hz
(Feedback divider output)
1/100-second counter
0
1
2
3
4
5
6
7
8
9
Approximate 10 Hz
(1/100-second counter output)
25/256 s
26/256 s
1/10-second counter count-up pattern
1/10-second counter
1 Hz
(1/10-second counter output)
26/256 × 6 + 25/256 × 4 = 1 s

Figure 16.2.1 Count-up Pattern of Stopwatch Timer

The feedback divider generates an approximate 100 Hz signal with 2/256-second and 3/256-second clock cycles from the 256 Hz signal supplied from the OSC module.

The 1/100-second counter counts the approximate 100 Hz signal output by the feedback divider and generates an approximate 10 Hz signal with 25/256-second and 26/256-second clock cycles. The counter counts up in 2/256-second and 3/256-second intervals, therefore the count cycle is an approximate 1/100 seconds.

The 1/10-second counter counts the approximate 10 Hz signal output by the 1/100-second counter in 4:6 ratios to generate a 1 Hz signal. The counter counts up in 25/256-second and 26/256-second intervals, therefore the count cycle is an approximate 1/10 seconds.

16.3 Operating Clock

The stopwatch timer uses the 256 Hz clock output from the OSC module as its operating clock.

The OSC module generates this clock by dividing the OSC1 clock by 128. Therefore, the clock frequency is 256 Hz when the OSC1 clock frequency is 32.768 kHz. Be aware that the frequencies described in this chapter change if the OSC1 clock has another frequency.

The OSC module does not provide a control bit for the 256 Hz clock. The 256 Hz clock is always supplied to the stopwatch timer when the OSC1 oscillator is on.

For control of the OSC1 oscillator, see Chapter 7, “Oscillator (OSC).”

16.4 Resetting the Stopwatch Timer

To reset the stopwatch timer, write 1 to the SWTRST bit (D4/SWT_CTL register). This initializes the counter to 0.

∗ **SWTRST**: Stopwatch Timer Reset Bit in the Stopwatch Timer Control (SWT_CTL) Register (D4/0x5020)

The counter is also initialized to 0 at initial reset.

16.5 Stopwatch Timer Run/Stop Control

Before starting the stopwatch timer, set up the conditions as shown below.
(1) Set up the interrupt level and enable the stopwatch timer interrupt if the interrupt is used. See Section 16.6.
(2) Reset the timer. See Section 16.4.

The stopwatch timer provides SWTRUN (D0/SWT_CTL register) to run and stop the counter.

* **SWTRUN**: Stopwatch Timer Run/Stop Control Bit in the Stopwatch Timer Control (SWT_CTL) Register (D0/0x5020)

The stopwatch timer starts counting when 1 is written to SWTRUN. The clock input is disabled and the timer stops counting when 0 is written to SWTRUN. This control does not affect the counter data (SWT_BCNT register). Even when the stopwatch timer has stopped counting, the counter retains its count so that the timer can start counting again from that point.
When both SWTRUN and SWTRST are set to 1 at the same time, the stopwatch timer starts counting after resetting the counter.

While the stopwatch timer is running, a cause of interrupt occurs at the falling edge of the 100 Hz (approximate 100 Hz), 10 Hz (approximate 10 Hz), and 1 Hz signals. If the interrupt has been enabled, an interrupt request is sent to the interrupt controller (ITC).

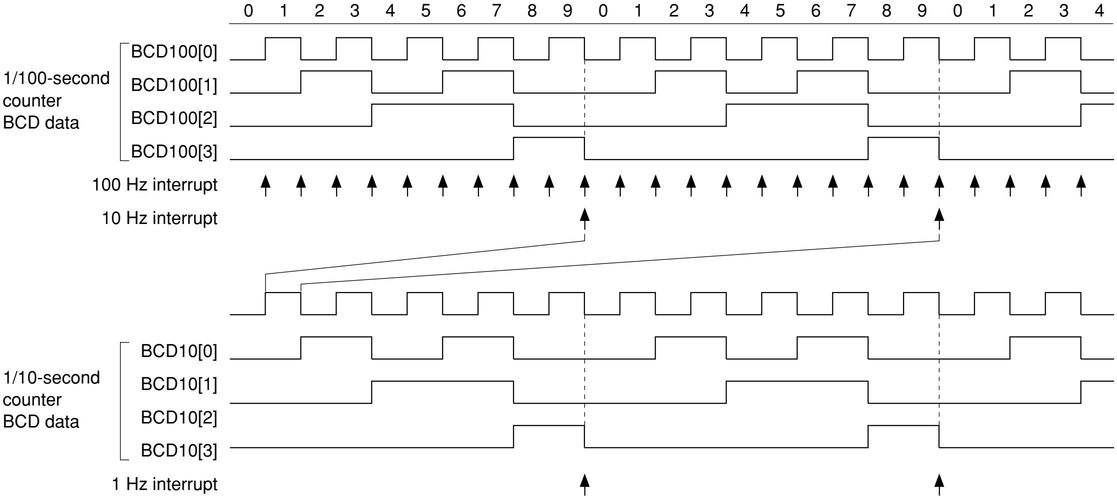


Figure 16.5.1 Timing Chart of Stopwatch Timer

Note: The Stopwatch timer is actually made to Run/Stop in sync with the falling edge of the 256 Hz signal after writing to SWTRUN. Therefore, when 0 is written to the SWTRUN, the timer stops after the counter is incremented. The SWTRUN maintains 1 for reading until the timer actually enters Stop status.
Figure 16.5.2 shows the timer operation at start/stop.

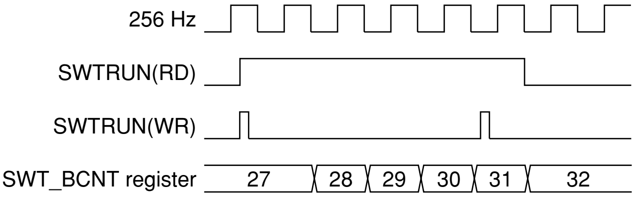


Figure 16.5.2 Stopwatch Timer Start/Stop Operation

16.6 Stopwatch Timer Interrupt

The SWT module can generate the following three types of interrupts:

- 100 Hz interrupt
- 10 Hz interrupt
- 1 Hz interrupt

The SWT module has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with the three causes of interrupt. To determine the cause of interrupt that has occurred, read the interrupt flags in the SWT module.

100 Hz, 10 Hz, and 1 Hz interrupts

An interrupt request occurs at the falling edge of the 100 Hz (approximate 100 Hz), 10 Hz (approximate 10 Hz), and 1 Hz signals, and it sets the interrupt flag in the SWT module to 1.

∗ **SIF1**: 1 Hz Interrupt Flag in the Stopwatch Timer Interrupt Flag (SWT_IFLG) Register (D2/0x5023)
∗ **SIF10**: 10 Hz Interrupt Flag in the Stopwatch Timer Interrupt Flag (SWT_IFLG) Register (D1/0x5023)
∗ **SIF100**: 100 Hz Interrupt Flag in the Stopwatch Timer Interrupt Flag (SWT_IFLG) Register (D0/0x5023)

Set the interrupt enable bit corresponding to the interrupt flag to 1 when using the interrupt. If the interrupt enable bit is set to 0 (default), the interrupt flag will not be set to 1 and an interrupt request by the corresponding signal will not be sent to the ITC.

∗ **SIE1**: 1 Hz Interrupt Enable Bit in the Stopwatch Timer Interrupt Mask (SWT_IMSK) Register (D2/0x5022)
∗ **SIE10**: 10 Hz Interrupt Enable Bit in the Stopwatch Timer Interrupt Mask (SWT_IMSK) Register (D1/0x5022)
∗ **SIE100**: 100 Hz Interrupt Enable Bit in the Stopwatch Timer Interrupt Mask (SWT_IMSK) Register (D0/0x5022)

If SIF∗ is set to 1, the SWT module outputs the interrupt request signal to the ITC. The interrupt request signal sets the stopwatch timer interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The stopwatch timer interrupt handler routine should read the SIF* flags to check the signal that causes occurrence of the interrupt.
Furthermore, the interrupt handler routine must reset (write 1 to) SIF∗ in the SWT module, not the stopwatch timer interrupt flag in the ITC, to clear the cause of interrupt.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the SIF∗ flags before the stopwatch timer interrupt is enabled using SIE∗.

ITC registers for stopwatch timer interrupt

The stopwatch timer asserts the interrupt signal sent to the ITC at the falling edge of the signal whose interrupt is enabled according to the interrupt condition settings shown above. To generate a stopwatch timer interrupt, set the interrupt level and enable the interrupt using the ITC registers.
The following shows the control bits for the stopwatch timer interrupt in the ITC.

Interrupt flag in the ITC

∗ **EIFT2**: Stopwatch Timer Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D2/0x4300)

Interrupt enable bit in the ITC

∗ **EIEN2**: Stopwatch Timer Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D2/0x4302)

Interrupt level setup bits in the ITC

∗ **EILV2[2:0]**: SWT Interrupt Level Bits in the External Interrupt Level Setup (ITC_ELV1) Register 1 (D[2:0]/0x4308)

Interrupt trigger mode select bit in the ITC (fixed at 1)

∗ **EITG2**: SWT Interrupt Trigger Mode Select Bit in the External Interrupt Level Setup (ITC_ELV1) Register 1 (D4/0x4308)

EIFT2 is set to 1 at the falling edge of the 100/10/1 Hz signal whose interrupt is enabled. If EIEN2 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the stopwatch timer interrupt, set EIEN2 to 0. EIFT2 is always set to 1 by the interrupt signal sent from the SWT module, regardless of how EIEN2 is set (even when set to 0).
EILV2[2:0] sets the interrupt level (0 to 7) of the stopwatch timer interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The stopwatch timer interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Chapter 6, "Interrupt Controller (ITC)."

Note: The settings shown below are required to manage the cause-of-interrupt occurrence status using the interrupt flags in the SWT module.

1. Set the stopwatch timer interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the SIF∗ interrupt flag of the SWT module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

Interrupt vector

The following shows the vector number and vector address for the stopwatch timer interrupt:

Vector number: 6 (0x06)
Vector address: 0x8018

16.7 Details of Control Registers

Table 16.7.1 List of Stopwatch Timer Registers

Address	Register name		Function
0x5020	SWT_CTL	Stopwatch Timer Control Register	Resets and starts/stops the timer.
0x5021	SWT_BCNT	Stopwatch Timer BCD Counter Register	BCD counter data
0x5022	SWT_IMSK	Stopwatch Timer Interrupt Mask Register	Enables/disables interrupt.
0x5023	SWT_IFLG	Stopwatch Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The following describes each stopwatch timer register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5020: Stopwatch Timer Control Register (SWT_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Stopwatch Timer Control Register (SWT_CTL)	**0x5020** (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	**SWTRST**	Stopwatch timer reset	1	Reset	0	Ignored	0	W	
		D3–1	–	reserved	–				–	–	
		D0	**SWTRUN**	Stopwatch timer run/stop control	1	Run	0	Stop	0	R/W	

D[7:5] Reserved

D4 SWTRST: Stopwatch Timer Reset Bit

Resets the stopwatch timer.

1 (W): Reset
0 (W): Has no effect
0 (R): Always 0 when read (default)

Writing 1 to this bit resets the counter to 0x0.
When the stopwatch timer is reset in Run state, it restarts immediately after resetting. In Stop state, the counter maintains 0x0.

D[3:1] Reserved

D0 SWTRUN: Stopwatch Timer Run/Stop Control Bit

Starts/stops the stopwatch timer.

1 (R/W): Run
0 (R/W): Stop (default)

The stopwatch timer starts counting by writing 1 to SWTRUN and stops by writing 0.
In Stop state, the counter data is retained until the timer is reset or placed in a Run state.

0x5021: Stopwatch Timer BCD Counter Register (SWT_BCNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer BCD Counter Register (SWT_BCNT)	**0x5021** (8 bits)	D7–4	**BCD10[3:0]**	1/10 sec. BCD counter value	0 to 9	0	R	
		D3–0	**BCD100[3:0]**	1/100 sec. BCD counter value	0 to 9	0	R	

D[7:4] BCD10[3:0]: 1/10 Sec. BCD Counter Value

The 1/10-second counter data can be read. (Default: 0)

This is a read-only register, so the writing operation is invalid.

D[3:0] BCD100[3:0]: 1/100 Sec. BCD Counter Value

The 1/100-second counter data can be read. (Default: 0)

This is a read-only register, so the writing operation is invalid.

Note: If this register is read while the counter is running, the read value may not represent the current counter value (an indefinite value may be read out).

The counter value should be obtained by one of the following procedures:

- Read the counter value after stopping the counter.
- Read the counter value twice to determine that both read results are the same and that the read value is significant.

0x5022: Stopwatch Timer Interrupt Mask Register (SWT_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Stopwatch Timer Interrupt Mask Register (SWT_IMSK)	**0x5022** (8 bits)	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**SIE1**	1 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**SIE10**	10 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**SIE100**	100 Hz interrupt enable	1	Enable	0	Disable	0	R/W	

This register enables or disables the interrupt requests by the stopwatch timer 100 Hz, 10 Hz, and 1 Hz signals, individually. Setting SIE∗ bit to 1 enables the stopwatch timer interrupt request by the falling edge of the corresponding signal; setting it to 0 disables the interrupt.

In addition, it is necessary to set the stopwatch timer interrupt enable bit in the ITC to interrupt enabled to actually generate an interrupt.

D[7:3] Reserved

D2 SIE1: 1 Hz Interrupt Enable Bit

Enables/disables the 1 Hz interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

D1 SIE10: 10 Hz Interrupt Enable Bit

Enables/disables the 10 Hz interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

D0 SIE100: 100 Hz Interrupt Enable Bit

Enables/disables the 100 Hz interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

0x5023: Stopwatch Timer Interrupt Flag Register (SWT_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Stopwatch Timer Interrupt Flag Register (SWT_IFLG)	**0x5023** (8 bits)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	**SIF1**	1 Hz interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	**SIF10**	10 Hz interrupt flag					0	R/W	
		D0	**SIF100**	100 Hz interrupt flag					0	R/W	

This register indicates the interrupt cause occurrence status by the stopwatch timer 100 Hz, 10 Hz, and 1 Hz signals. When a stopwatch timer interrupt occurs, read the interrupt flag in this register to determine the cause of interrupt that has occurred (or frequency).

The SIF∗ bits are the interrupt flags that correspond to 100 Hz, 10 Hz, and 1 Hz interrupts respectively, and are set to 1 at the falling edge of the signals if the corresponding SIE∗ bit (SWT_IMSK register) has been set to 1. At this time, the stopwatch timer interrupt request signal is output to the ITC. The interrupt request signal sets the stopwatch timer interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

The settings shown below are required to manage the cause-of-interrupt occurrence status using this register.

1. Set the stopwatch timer interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the interrupt flag of the SWT module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

The SIF∗ flags are reset by writing 1.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the SIF∗ flags before the stopwatch timer interrupt is enabled using SIE∗.

D[7:3] Reserved

D2 SIF1: 1 Hz Interrupt Flag

This is the interrupt flag to indicate the 1 Hz interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

SIF1 is set to 1 at the falling edge of the 1 Hz signal only when SIE1 (D2/SWT_IMSK register) has been set to 1.

D1 SIF10: 10 Hz Interrupt Flag

This is the interrupt flag to indicate the 10 Hz interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

SIF10 is set to 1 at the falling edge of the 10 Hz signal only when SIE10 (D1/SWT_IMSK register) has been set to 1.

D0 SIF100: 100 Hz Interrupt Flag

This is the interrupt flag to indicate the 100 Hz interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

SIF100 is set to 1 at the falling edge of the 100 Hz signal only when SIE100 (D0/SWT_IMSK register) has been set to 1.

16.8 Precautions

- Before the stopwatch timer can start counting, the OSC1 oscillator must be turned on.
- To avoid occurrence of unnecessary interrupts, be sure to reset the interrupt flags in the SWT_IFLG register before the stopwatch timer interrupt is enabled using the SWT_IMSK register.
- The Stopwatch timer is actually made to Run/Stop in sync with the falling edge of the 256 Hz signal after writing to SWTRUN. Therefore, when 0 is written to the SWTRUN, the timer stops after the counter is incremented. The SWTRUN maintains 1 for reading until the timer actually enters Stop status.
  Figure 16.8.1 shows the timer operation at start/stop.

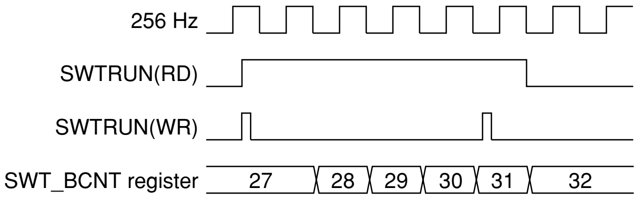


Figure 16.8.1 Stopwatch Timer Start/Stop Operation

- If the `slp` instruction is executed while the stopwatch timer is running (SWTRUN = 1), the stopwatch timer will be unstable immediately after SLEEP status is canceled. Therefore, the stopwatch timer should be stopped (SWTRUN = 0) before the `slp` instruction is executed to enter SLEEP mode.
- If the counter register is read while the counter is running, the read value may not represent the current counter value (an indefinite value may be read out).
  To obtain the counter value, read the counter register after stopping the counter. Or read the counter value twice to determine that both read results are the same and that the read value is significant.

THIS PAGE IS BLANK.

17 Watchdog Timer (WDT)

17.1 Outline of the Watchdog Timer

The S1C17701 is equipped with a watchdog timer driven by OSC1 as the clock source.

If the watchdog timer is not reset for more than 131072/fosc1 seconds (4 seconds when fosc1 = 32.768 kHz), an NMI or a reset signal (selectable with software) is generated and output to the CPU. By programming the watchdog timer to be reset within this period so that an NMI/reset will not occur, it is possible to detect program runaway as if the processing has not been executed.

Figure 17.1.1 shows the structure of the watchdog timer.

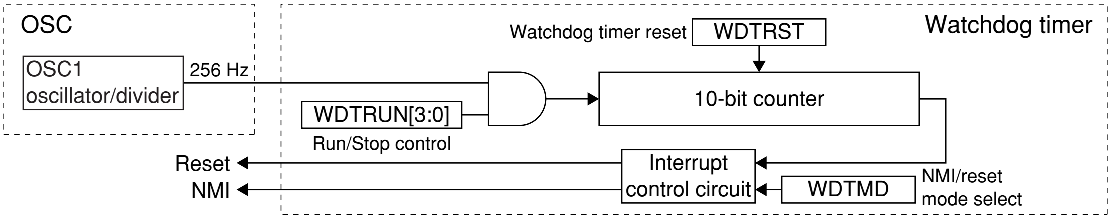


Figure 17.1.1 Structure of Watchdog Timer

17.2 Operating Clock

The watchdog timer uses the 256 Hz clock output from the OSC module as its operating clock.

The OSC module generates this clock by dividing the OSC1 clock by 128. Therefore, the clock frequency is 256 Hz when the OSC1 clock frequency is 32.768 kHz. Be aware that the frequencies and times described in this chapter change if the OSC1 clock has another frequency.

The OSC module does not provide a control bit for the 256 Hz clock. The 256 Hz clock is always supplied to the watchdog timer when the OSC1 oscillator is on.

For control of the OSC1 oscillator, see Chapter 7, “Oscillator (OSC).”

17.3 Controlling the Watchdog Timer

17.3.1 Selecting NMI/Reset Mode

The watchdog timer allows selection whether it outputs an NMI signal or a reset signal when the watchdog timer is not reset within the NMI/reset generating cycle. Use WDTMD (D1/WDT_ST register) for this selection.

∗ **WDTMD**: NMI/Reset Mode Select Bit in the Watchdog Timer Status (WDT_ST) Register (D1/0x5041)

Set WDTMD to 0 (default) to generate an NMI or set it to 1 to generate a reset.

17.3.2 Watchdog Timer Run/Stop Control

Writing a value other than 0b1010 to WDTRUN[3:0] (D[3:0]/WDT_CTL register) starts counting by the watchdog timer; writing 0b1010 stops the watchdog timer.

∗ **WDTRUN[3:0]**: Watchdog Timer Run/Stop Control Bits in the Watchdog Timer Control (WDT_CTL) Register (D[3:0]/0x5040)

At initial reset, WDTRUN[3:0] is set to 0b1010 and the watchdog timer does not start counting.
Depending on the counter value while the watchdog timer is idle, an NMI/reset may occur immediately after the watchdog timer starts. Therefore, when starting the watchdog timer, reset it at the same time (as described in the next section).

17.3.3 Resetting the Watchdog Timer

To reset the watchdog timer, write 1 to WDTRST (D4/WDT_CTL register).

∗ **WDTRST**: Watchdog Timer Reset Bit in the Watchdog Timer Control (WDT_CTL) Register (D4/0x5040)

When using the watchdog timer, prepare a routine to reset the watchdog timer before generating an NMI or reset in a location to be processed periodically. Make sure that this routine is processed within 131072/f_{OSC1} second cycles (4 seconds when f_{OSC1} = 32.768 kHz). After the watchdog counter is reset, it starts counting NMI/reset generation cycles all over again.

If the watchdog timer is not reset within the NMI/reset generation cycle for some reason, the CPU is placed into interrupt handling by an NMI or reset signal. In the interrupt handling, the CPU reads the interrupt vector and executes the handler routine. The reset and NMI vector addresses are set to 0x8000 and 0x8008, respectively.

When the watchdog timer is not reset and an NMI occurs due to a counter overflow, WDTST (D0/WDT_ST register) is set to 1.

∗ **WDTST**: NMI Status Bit in the Watchdog Timer Status (WDT_ST) Register (D0/0x5041)

This bit is provided to check if an NMI has occurred by the watchdog timer.
WDTST being set to 1 can be cleared to 0 by resetting the watchdog timer.

17.3.4 Operation in Standby Mode

In HALT mode

In HALT mode, the watchdog timer remains active as its operating clock is supplied. Therefore, if HALT mode remains active beyond the NMI/reset generation cycle, an NMI or reset signal deactivates HALT mode.
To disable the watchdog timer in HALT mode, write 0b1010 to WDTRUN[3:0] to stop the watchdog timer before executing the `halt` instruction. In this case, be sure to reset the watchdog timer before restarting after HALT mode is canceled.

In SLEEP mode

The clock supply from the OSC module stops in SLEEP mode. Therefore, the watchdog timer also stops operating. To prevent an unnecessary NMI or reset signal from being generated after canceling SLEEP mode, be sure to reset the watchdog timer before executing the `slp` instruction. Moreover, stop the watchdog timer by setting WDTRUN[3:0] as required.

17.4 Details of Control Registers

Table 17.4.1 List of Watchdog Timer Registers

Address	Register name		Function
0x5040	WDT_CTL	Watchdog Timer Control Register	Resets and starts/stops the timer.
0x5041	WDT_ST	Watchdog Timer Status Register	Sets the timer mode and indicates NMI status.

The following describes each watchdog timer register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all "reserved bits."

0x5040: Watchdog Timer Control Register (WDT_CTL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Watchdog Timer Control Register (WDT_CTL)	0x5040 (8 bits)	D7–5	–	reserved	–		–	–	0 when being read.
		D4	WDTRST	Watchdog timer reset	1 Reset	0 Ignored	0	W	
		D3–0	WDTRUN[3:0]	Watchdog timer run/stop control	Other than 1010 Run	1010 Stop	1010	R/W	

D[7:5] Reserved

D4 WDTRST: Watchdog Timer Reset Bit

Resets the watchdog timer.

1 (W): Reset

0 (W): Has no effect

0 (R): Always 0 when read (default)

When the watchdog timer is used, it must be reset by writing 1 to this bit within the NMI/reset generation cycle (4 seconds when f_{OSC1} = 32.768 kHz). The up-counter is thereby reset to 0, then starts counting NMI/reset generation cycles all over again.

D[3:0] WDTRUN[3:0]: Watchdog Timer Run/Stop Control Bits

Starts/stops the watchdog timer.

Other than 0b1010 (R/W): Run

0b1010 (R/W): Stop (default)

Reset the watchdog timer before starting, thus preventing the generation of unnecessary NMI or reset signals.

0x5041: Watchdog Timer Status Register (WDT_ST)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Watchdog Timer Status Register (WDT_ST)	**0x5041** (8 bits)	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**WDTMD**	NMI/Reset mode select	1	Reset	0	NMI	0	R/W	
		D0	**WDTST**	NMI status	1	NMI occurred	0	Not occurred	0	R	

D[7:2] Reserved

D1 WDTMD: NMI/Reset Mode Select Bit

Selects either NMI or reset to be generated when the counter overflows.

1 (R/W): Reset

0 (R/W): NMI (default)

When this bit is set to 1, a reset signal will be output when the counter overflows. When this bit is set to 0, an NMI signal will be output.

D0 WDTST: NMI Status Bit

Indicates that an NMI has occurred due to a counter overflow.

1 (R): NMI has occurred (counter overflowed)

0 (R): NMI has not occurred (default)

This bit is provided to check if an NMI has occurred by the watchdog timer. WDTST being set to 1 can be cleared to 0 by resetting the watchdog timer.

This bit is also set due to a counter overflow even if reset output is selected, however, the bit is cleared at initial reset and cannot be checked if it has been set to 1.

17.5 Precautions

- When the watchdog timer is used, it must be reset within 131072/f_{OSC1} second cycle (4 seconds when f_{OSC1} = 32.768 kHz).
- Reset the watchdog timer before starting, thus preventing the generation of unnecessary NMI or reset signals.

THIS PAGE IS BLANK.

18 UART

18.1 Outline of the UART

The S1C17701 equipped with one channel of UART. The UART performs asynchronous data transfer from/to an external serial device in a 150 to 115200 bps transfer rate. The UART contains two-byte receive data buffer and one-byte transmit data buffer allowing full-duplex communication. The transfer clock is internally generated using a timer module or an external clock is input from the SCLK (P25) pin. The character length (seven or eight bits), number of stop bits (one or two bits), and parity mode (even, odd, or none) are programmable. The start bit is fixed at one bit. In data receive operation, overrun, framing, and parity errors are detectable. The UART can generate three types of interrupts (transmit buffer empty, receive buffer full, and receive error), this makes it possible to process serial data transfer simply in an interrupt handler.

Furthermore, the UART module contains an RZI modulator/demodulator, allowing an infrared-ray communication circuit to be configured based on IrDA 1.0 simply by adding an external circuit.

Figure 18.1.1 shows the structure of the UART.

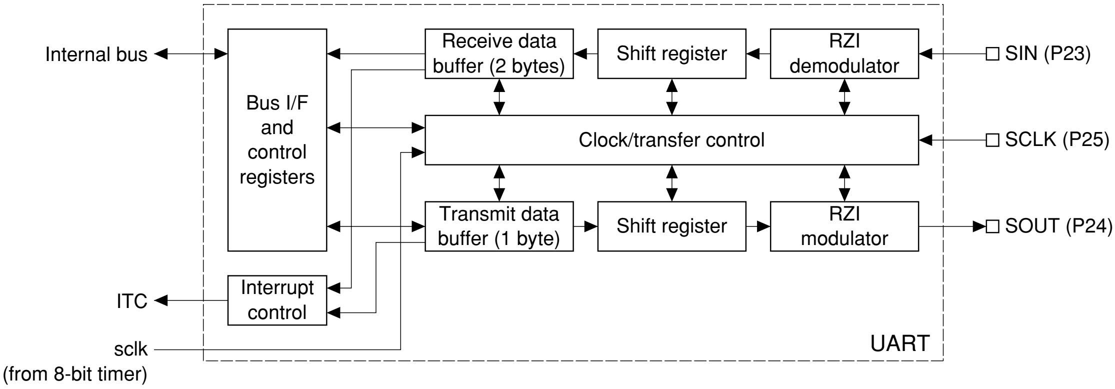


Figure 18.1.1 Structure of UART

18.2 UART Pins

Table 18.2.1 lists the I/O pins for the UART.

Table 18.2.1 List of UART Pins

Pin name	I/O	Size	Function
SIN (P23)	I	1	UART data input pin This pin inputs serial data sent from an external serial device.
SOUT (P24)	O	1	UART data output pin This pin outputs serial data to be sent to an external serial device.
SCLK (P25)	I	1	UART clock input pin This pin inputs the transfer clock when an external clock is used.

The UART input/output pins (SIN, SOUT, SCLK) are shared with the I/O ports (P23, P24, P25) and they are initialized as general-purpose I/O port pins by default. Before using these pins for the UART, the pin functions must be switched using the P2_PMUX register. Set the control bits shown below to 1 to configure the pins for the serial interface.

P23 → SIN

∗ **P23MUX**: P23 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D3/0x52a2)

P24 → SOUT

∗ **P24MUX**: P24 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D4/0x52a2)

P25 → SCLK (required only when an external clock is used)

∗ **P25MUX**: P25 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D5/0x52a2)

For details on switching pin function, see Section 10.2, “Selecting I/O Pin Functions (Port MUX).”

18.3 Transfer Clock

The UART allows the application to select either the internal clock or an external clock as the transfer clock. Use the SSCK bit (D0/UART_MOD register) for this selection.

∗ **SSCK**: Input Clock Select Bit in the UART Mode (UART_MOD) Register (D0/0x4103)

Note: Make sure that the UART is disabled (RXEN/UART_CTL register = 0) when alter the SSCK bit.

∗ **RXEN**: UART Enable Bit in the UART Control (UART_CTL) Register (D0/0x4104)

Internal clock

When SSCK is set to 0 (default), the internal clock is selected. The UART uses the 8-bit timer output clock as the transfer clock. Therefore, it is necessary to program the 8-bit timer so that it will output a clock according to the transfer rate.
See Chapter 12, "8-bit Timer (T8F)," for controlling the 8-bit timer.

External clock

When SSCK is set to 1, an external clock is selected. Configure the P25 pin as the SCLK pin (see Section 18.2) and input an external clock to the pin.

Notes:
- The UART divides the 8-bit timer output clock or external clock by 16 to generate the sampling clock. Make sure of the division ratio when setting a transfer rate.
- The frequency of the external clock input from the SCLK pin must be half of PCLK or lower and the clock duty ratio must be 50%.

18.4 Setting Transfer Data Conditions

The following conditions are selectable to configure transfer data format:

- Character length: 7 or 8 bits
- Start bit: 1 bit, fixed
- Stop bit: 1 or 2 bits
- Parity bit: Even, odd, or none

Note: Make sure that the UART is disabled (RXEN/UART_CTL register = 0) when setting the transfer data conditions.

∗ **RXEN**: UART Enable Bit in the UART Control (UART_CTL) Register (D0/0x4104)

Character length

Use the CHLN bit (D4/UART_MOD register) to select the character length. When CHLN is set to 0 (default), the character length is configured to seven bits; when CHLN is set to 1, the character length is configured to eight bits.

∗ **CHLN**: Character Length Select Bit in the UART Mode (UART_MOD) Register (D4/0x4103)

Stop bit

Use the STPB bit (D1/UART_MOD register) to select the stop bit length. When STPB is set to 0 (default), the stop bit length is set to one bit; when STPB is set to 1, the stop bit length is set to two bits.

∗ **STPB**: Stop Bit Select Bit in the UART Mode (UART_MOD) Register (D1/0x4103)

Parity bit

Use the PREN bit (D3/UART_MOD register) to select whether the parity function is enabled or not. When PREN is set to 0 (default), parity function is disabled. In this case, a parity bit will not be added to transfer data and the parity check will not be performed when data is received. When PREN is set to 1, parity function is enabled. In this case, a parity bit will be added to transfer data and the parity check will be performed when data is received.

When the parity function is enabled, select a parity mode using the PMD bit (D2/UART_MOD register). When PMD is set to 0 (default), the parity bit is added/checked as even parity; when PMD is set to1, the parity bit is added/checked as odd parity.

∗ **PREN**: Parity Enable Bit in the UART Mode (UART_MOD) Register (D3/0x4103)
∗ **PMD**: Parity Mode Select Bit in the UART Mode (UART_MOD) Register (D2/0x4103)

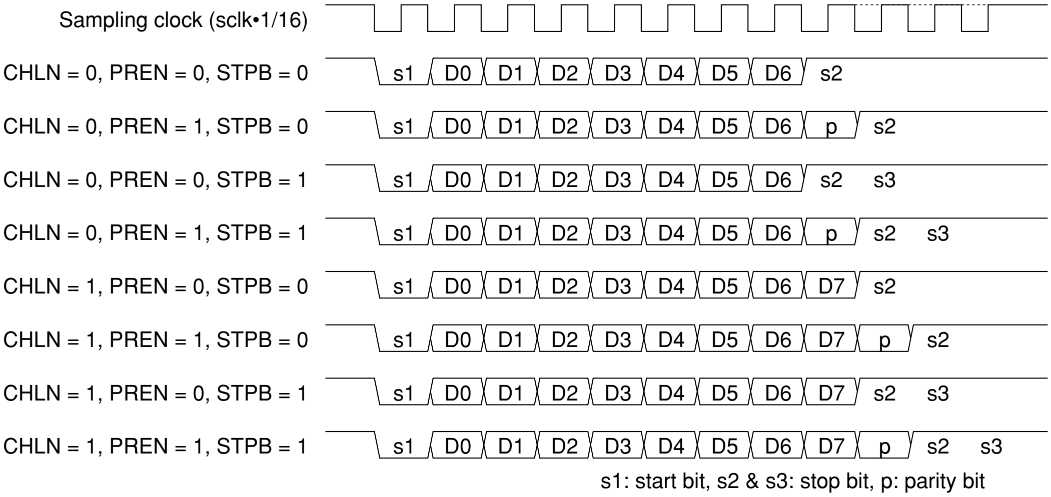

Sampling clock (sclk•1/16)
CHLN = 0, PREN = 0, STPB = 0
s1 D0 D1 D2 D3 D4 D5 D6 s2
CHLN = 0, PREN = 1, STPB = 0
s1 D0 D1 D2 D3 D4 D5 D6 p s2
CHLN = 0, PREN = 0, STPB = 1
s1 D0 D1 D2 D3 D4 D5 D6 s2 s3
CHLN = 0, PREN = 1, STPB = 1
s1 D0 D1 D2 D3 D4 D5 D6 p s2 s3
CHLN = 1, PREN = 0, STPB = 0
s1 D0 D1 D2 D3 D4 D5 D6 D7 s2
CHLN = 1, PREN = 1, STPB = 0
s1 D0 D1 D2 D3 D4 D5 D6 D7 p s2
CHLN = 1, PREN = 0, STPB = 1
s1 D0 D1 D2 D3 D4 D5 D6 D7 s2 s3
CHLN = 1, PREN = 1, STPB = 1
s1 D0 D1 D2 D3 D4 D5 D6 D7 p s2 s3
s1: start bit, s2 & s3: stop bit, p: parity bit

Figure 18.4.1 Transfer Data Format

18.5 Data Transmit/Receive Control

Before starting data transfer, set up the conditions as shown below.

(1) Select an input clock. See Section 18.3.
Set up the 8-bit timer to output the transfer clock if the internal clock is used as the transfer clock. See Chapter 12.

(2) Configure the transfer data format. See Section 18.4.

(3) Set IrDA mode when using the IrDA interface. See Section 18.8.

(4) Set up the interrupt conditions if the UART interrupt is used. See Section 18.7.

Note: Make sure that the UART is disabled (RXEN/UART_CTL register = 0) when setting the conditions above.

* **RXEN**: UART Enable Bit in the UART Control (UART_CTL) Register (D0/0x4104)

Enabling data transmission/reception

First, set the RXEN bit (D0/UART_CTL register) to 1 to enable data transmission/reception. This puts the transmitter/receiver in ready-to-transmit/receive status.

Note: Do not set the RXEN bit to 0 while the UART is transmitting/receiving data.

Data transmit control

To start transmission, write transmit data to the UART_TXD register (0x4101).

* **UART_TXD**: UART Transmit Data Register (0x4101)

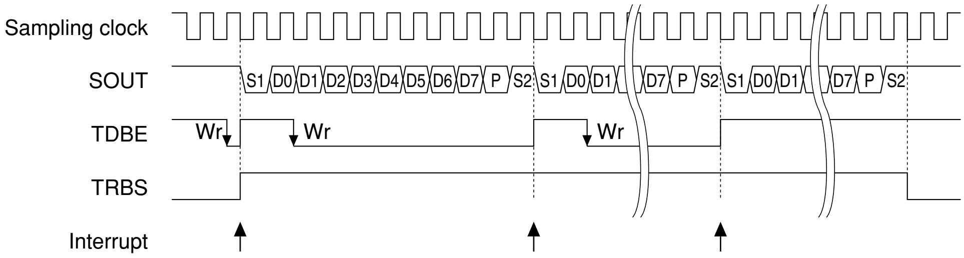
Data is written to the transmit data buffer and the transmitter starts data transmission.
The buffered data is sent to the shift register for transmission and a start bit is output from the SOUT pin. Then data in the shift register is output from the LSB. The transmit data bits are shifted in sync with the rising edge of the sampling clock and output from the SOUT pin sequentially. After the MSB has been output, a parity bit (if parity is enabled) and a stop bit are output.

The transmitter provides two status flags, TDBE (D0/UART_ST register) and TRBS (D2/UART_ST register).

* **TDBE**: Transmit Data Buffer Empty Flag in the UART Status (UART_ST) Register (D0/0x4100)
* **TRBS**: Transmit Busy Flag in the UART Status (UART_ST) Register (D2/0x4100)

The TDBE flag indicates the transmit data buffer status; it goes 0 when the application program writes data to the transmit data buffer and returns to 1 when the data in the transmit data buffer is sent to the shift register for transmitting. An interrupt can be generated when this flag goes 1 (see Section 18.7). Use this interrupt or read the TDBE flag to check that the transmit data buffer is empty before transmitting the next data. Although the transmit data buffer size is one byte, transmit data can be written while the previous data is being transmitted as the shift register is separately provided. However, make sure that the transmit data buffer is empty before writing transmit data. If data is written when the TDBE flag is 0, the previous transmit data in the transmit data buffer is overwritten with the new data.

The TRBS flag indicates the shift register status; it goes 1 when transmit data is loaded from the transmit data buffer and returns to 0 upon completion of a data transmission. Read this flag to check whether the transmitter is busy or idle.



S1: Start bit, S2: Stop bit, P: Parity bit, Wr: Data write to transmit data buffer

Figure 18.5.1 Data Transmit Timing Chart

Data receive control

The receiver activates by setting the RXEN bit to 1 and is ready to receive data sent from an external serial device.
When an external serial device has sent a start bit, the receiver detects its low level and starts following data bit sampling. The data bits are sampled at the rising edge of the sampling clock and received in the receive shift register assuming that the first data bit is LSB. After the MSB is received in the shift register, the received data is loaded to the receive data buffer. At the same time, the receiver performs a parity check with the parity bit received after the MSB if parity check is enabled.
The receive data buffer is a two-byte FIFO and can receive data until it becomes full.

The received data in the buffer can be read from the UART_RXD register (0x4102). The older data is read out first and cleared by reading.

∗ **UART_RXD**: UART Receive Data Register (0x4102)

The receiver provides two buffer status flags, RDRY (D1/UART_ST register) and RD2B (D3/UART_ST register).

∗ **RDRY**: Receive Data Ready Flag in the UART Status (UART_ST) Register (D1/0x4100)
∗ **RD2B**: Second Byte Receive Flag in the UART Status (UART_ST) Register (D3/0x4100)

The RDRY flag indicates that the receive data buffer contains the received data. The RD2B flag indicates that the receive data buffer is full.

(1) RDRY = 0, RD2B = 0
No data has been received. Therefore, it is not necessary to read the receive data buffer.

(2) RDRY = 1, RD2B = 0
One data has been received. Read the receive data buffer once. This reading clears the read data and resets the RDRY flag. The buffer status returns to (1) above.
If the receive data buffer is read twice, the second read value is invalid data.

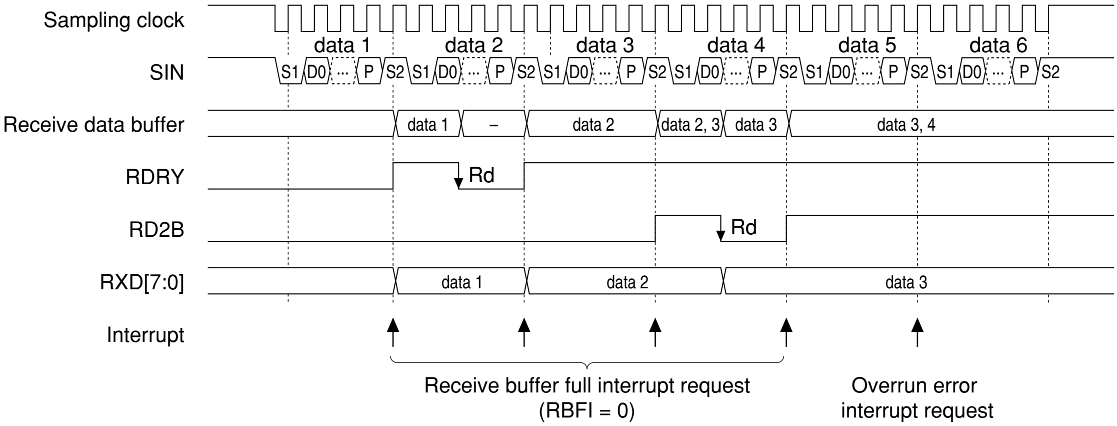
(3) RDRY = 1, RD2B = 1
Two data have been received. Read the receive data buffer twice. The receive data buffer outputs the older received data in the first reading. This reading clears the read data and resets the RD2B flag. The buffer status goes to (2) above. The latest received data is output in the second reading. The buffer status returns to (1) above after reading twice.
The shift register can receive one more data even if the receive data buffer is full. If an additional data is sent from the external serial device in this status, an overrun error occurs and the data in the shift register is overwritten with the new data. Therefore, be sure to read the receive data buffer before an overrun error occurs. Refer to Section 18.6 for the overrun error.

By reading these flags, the application program can check how many data have been received.

Furthermore, the UART can generate a receive data buffer full interrupt when data is received in the receive data buffer. This interrupt can be used to read the received data. A receive data buffer full interrupt occurs when one data has been received in the receive data buffer (status (2) above) by default. This may be changed by setting the RBFI bit (D1/UART_CTL register) to 1 so that the interrupt will occur when two data have been received in the received data buffer.

∗ **RBFI**: Receive Buffer Full Interrupt Condition Setup Bit in the UART Control (UART_CTL) Register (D1/0x4104)

In addition to the flags above, three receive error flags are provided. Refer to Section 18.6 for these flags and details of receive errors.



S1: Start bit, S2: Stop bit, P: Parity bit, Rd: Data read from RXD[7:0]

Figure 18.5.2 Data Receive Timing Chart

Disabling data transmission/reception

After data transfer (both transmission and reception) has finished, write 0 to the RXEN bit to disable data transmission/reception.

Always make sure that the TDBE flag is 1 and TRBS and RDRY flags are 0 before data transmission/reception is disabled.

When the RXEN bit is set to 0, the transmit and receive data buffers are placed in empty status (data is cleared if any remains). Furthermore, the data being transferred cannot be guaranteed if RXEN is set to 0 during transmitting/receiving.

18.6 Receive Errors

Three types of receive errors can be detected in data reception.

The receive errors are causes of interrupt, so the error can be processed in the interrupt handler routine. Refer to Section 18.7 for controlling the UART interrupts.

Parity error

If the PREN bit (D3/UART_MOD register) is set to 1 (parity enabled), the parity bit is checked when data is received.

This parity check is performed when the data received in the shift register is loaded to the receive data buffer in order to check conformity with the PMD bit (D2/UART_MOD register) setting (odd or even parity).

If any nonconformity is found in this check, a parity error is assumed and the parity error flag PER (D5/UART_ST register) is set to 1.

Even when this error occurs, the received data in error is loaded to the receive data buffer and the receive operation is continued. However, the received data in which a parity error has occurred cannot be guaranteed.

The PER flag (D5/UART_ST register) is reset to 0 by writing 1.

∗ **PREN**: Parity Enable Bit in the UART Mode (UART_MOD) Register (D3/0x4103)
∗ **PMD**: Parity Mode Select Bit in the UART Mode (UART_MOD) Register (D2/0x4103)
∗ **PER**: Parity Error Flag in the UART Status (UART_ST) Register (D5/0x4100)

Framing error

If data with a stop bit = 0 is received, the UART assumes that the data is out of sync and generates a framing error.

If two stop bits are used, only the first stop bit is checked.

When this error occurs, the framing-error flag FER (D6/UART_ST register) is set to 1.

Even when this error occurs, the received data in error is loaded to the receive data buffer and the receive operation is continued. However, the received data in which a framing error has occurred cannot be guaranteed, even if no framing error is found in the following data received.

The FER flag (D6/UART_ST register) is reset to 0 by writing 1.

∗ **FER**: Framing Error Flag in the UART Status (UART_ST) Register (D6/0x4100)

Overrun error

Even when the receive data buffer is full (two data have been received), the next (third) data can be received into the shift register. If there is no space in the buffer (data has not been read) when the third data has been received, the third data in the shift register cannot be transferred to the buffer. If one more (fourth) data is transferred to this UART, the shift register (third data) is overwritten with the fourth data and an overrun error occurs.

When an overrun error occurs, the overrun error flag OER (D4/UART_ST register) is set to 1.

Even when this error occurs, the receive operation is continued.

The OER flag (D4/UART_ST register) is reset to 0 by writing 1.

∗ **OER**: Overrun Error Flag in the UART Status (UART_ST) Register (D4/0x4100)

18.7 UART Interrupt

The UART can generate the following three types of interrupts:

- Transmit buffer empty interrupt
- Receive buffer full interrupt
- Receive error interrupt

The UART has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with all three causes of interrupt. To determine the cause of interrupt that has occurred, read the status and error flags.

Transmit buffer empty interrupt

Set the TIEN bit (D4/UART_CTL register) to 1 when using this interrupt. If TIEN is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **TIEN**: Transmit Buffer Empty Interrupt Enable Bit in the UART Control (UART_CTL) Register (D4/0x4104)

When the transmit data set in the transmit data buffer is transferred to the shift register, the UART sets the TDBE bit (D0/UART_ST register) to 1 to indicate that the transmit data buffer is empty. At the same time, the UART outputs an interrupt request pulse to the ITC if the transmit buffer empty interrupt has been enabled (TIEN = 1).

∗ **TDBE**: Transmit Data Buffer Empty Flag in the UART Status (UART_ST) Register (D0/0x4100)

If other interrupt conditions are satisfied, an interrupt is generated.
The UART interrupt handler routine should read the TDBE flag to check if the interrupt has occurred due to a transmit buffer empty or another cause. When TDBE = 1, the UART interrupt handler routine can write the next transmit data to the transmit data buffer.

Receive buffer full interrupt

Set the RIEN bit (D5/UART_CTL register) to 1 when using this interrupt. If RIEN is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **RIEN**: Receive Buffer Full Interrupt Enable Bit in the UART Control (UART_CTL) Register (D5/0x4104)

When the specified number of received data is loaded to the receive data buffer, the UART outputs an interrupt request pulse to the ITC if the receive buffer full interrupt has been enabled (RIEN = 1). If the RBFI bit (D1/UART_CTL register) is 0, an interrupt request pulse is output when received data is loaded to the receive data buffer (when the RDRY flag (D1/UART_ST register) goes 1). If the RBFI bit (D1/UART_CTL register) is 1, an interrupt request pulse is output when two received data occupy the receive data buffer (when the RD2B flag (D3/UART_ST register) goes 1).

∗ **RBFI**: Receive Buffer Full Interrupt Condition Setup Bit in the UART Control (UART_CTL) Register (D1/0x4104)
∗ **RDRY**: Receive Data Ready Flag in the UART Status (UART_ST) Register (D1/0x4100)
∗ **RD2B**: Second Byte Receive Flag in the UART Status (UART_ST) Register (D3/0x4100)

If other interrupt conditions are satisfied, an interrupt is generated.
The UART interrupt handler routine should read the RDRY and RD2B flags to check if the interrupt has occurred due to a receive buffer full or another cause. When RDRY or RD2B = 1, the UART interrupt handler routine can read the received data from the receive data buffer.

Receive error interrupt

Set the REIEN bit (D6/UART_CTL register) to 1 when using this interrupt. If REIEN is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **REIEN**: Receive Error Interrupt Enable Bit in the UART Control (UART_CTL) Register (D6/0x4104)

When a parity, framing, or overrun error is detected during data reception, the UART sets the error flag listed below to 1 and outputs an interrupt request pulse to the ITC if the receive error interrupt has been enabled (REIEN = 1).

∗ **PER**: Parity Error Flag in the UART Status (UART_ST) Register (D5/0x4100)
∗ **FER**: Framing Error Flag in the UART Status (UART_ST) Register (D6/0x4100)
∗ **OER**: Overrun Error Flag in the UART Status (UART_ST) Register (D4/0x4100)

If other interrupt conditions are satisfied, an interrupt is generated.
The UART interrupt handler routine should read the error flags to check if the interrupt has occurred due to a receive error or another cause. When an error flag has been set to 1, the UART interrupt handler routine should execute an error recovery process.

ITC registers for UART interrupts

The following shows the control bits of the ITC provided for the UART:

Interrupt flag

∗ **IIFT4**: UART Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D12/0x4300)

Interrupt enable bits

∗ **IIEN4**: UART Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D12/0x4302)

Interrupt level setup bits

∗ **IILV4[2:0]**: UART Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV2) Register 2 (D[2:0]/0x4312)

When the UART outputs an interrupt request pulse, the corresponding interrupt flag is set to 1.
If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the UART interrupt, set the interrupt enable bit to 0.
The interrupt flag is always set to 1 by the UART interrupt request pulse, regardless of how the interrupt enable register is set (even when set to 0).
The interrupt level setup bits set the interrupt level (0 to 7) of the UART interrupt.
An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The UART interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Chapter 6, “Interrupt Controller (ITC).”

Interrupt vector

The following shows the vector number and vector address for the UART interrupt:

Vector number: 16 (0x10)
Vector address: 0x8040

18.8 IrDA Interface

The UART module contains an RZI modulator/demodulator, allowing an infrared-ray communication circuit to be configured based on IrDA 1.0 simply by adding an external circuit.

The transmit data output from the shift register of the UART is input to the modulator to convert the low pulse width into 3/16 sclk cycles before it is output from the SOUT pin.

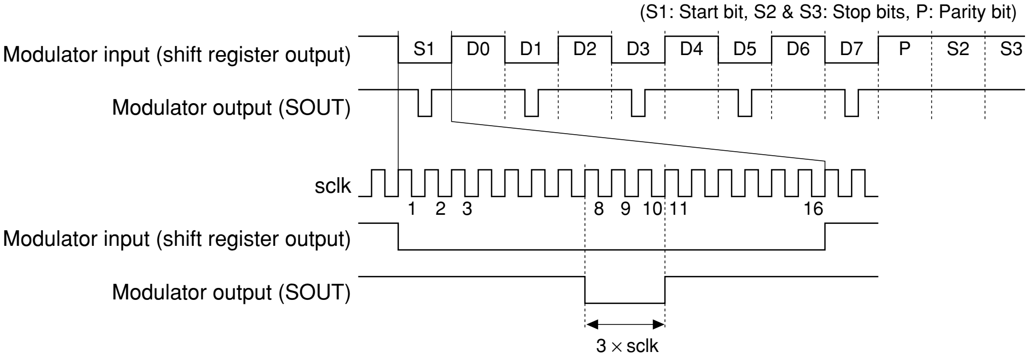

(S1: Start bit, S2 & S3: Stop bits, P: Parity bit)
Modulator input (shift register output)
S1
D0
D1
D2
D3
D4
D5
D6
D7
P
S2
S3
Modulator output (SOUT)
sclk
1
2
3
8
9
10
11
16
Modulator input (shift register output)
Modulator output (SOUT)
3 × sclk

Figure 18.8.1 Transmit Signal Waveform

The received IrDA signal is input to the demodulator to convert the low pulse width into 16 sclk cycles before input to the shift register for receiving. To detect low pulses input to the demodulator (minimum pulse width = 1.41 μs at 115200 bps), the demodulator uses a pulse detection clock selected from the prescaler output clocks separately with the transfer clock sclk.

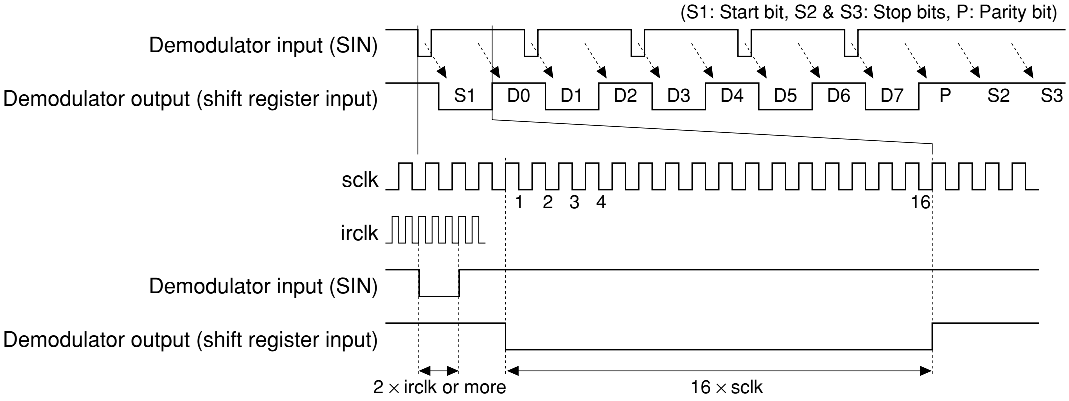

(S1: Start bit, S2 & S3: Stop bits, P: Parity bit)
Demodulator input (SIN)
Demodulator output (shift register input)
S1
D0
D1
D2
D3
D4
D5
D6
D7
P
S2
S3
sclk
1
2
3
4
16
irclk
Demodulator input (SIN)
Demodulator output (shift register input)
2 × irclk or more
16 × sclk

Figure 18.8.2 Receive Signal Waveform

Enabling the IrDA mode

To use the IrDA interface function, set the IRMD bit (D0/UART_EXP register) to 1. This enables the RZI modulator/demodulator.

∗ **IRMD**: IrDA Mode Select Bit in the UART Expansion (UART_EXP) Register (D0/0x4105)

Note: This setting must be performed before setting other UART conditions.

Selecting the IrDA receive detection clock

Select a prescaler output clock within the range from PCLK•1/1 to PCLK•1/128 as the input pulse detection clock using the IRCLK[2:0] bits (D[6:4]/UART_EXP register).

∗ **IRCLK[2:0]**: IrDA Receive Detection Clock Select Bits in the UART Expansion (UART_EXP) Register (D[6:4]/0x4105)

Table 18.8.1 Selecting the IrDA Receive Detection Clock

IRCLK[2:0]	Prescaler output clock
0x7	PCLK•1/128
0x6	PCLK•1/64
0x5	PCLK•1/32
0x4	PCLK•1/16
0x3	PCLK•1/8
0x2	PCLK•1/4
0x1	PCLK•1/2
0x0	PCLK•1/1

(Default: 0x0)

This clock must be faster than the transfer clock sclk supplied from the 8-bit timer or input from the SCLK pin. The demodulator regards a low pulse of which the width is longer than two cycles of the IrDA receive detection clock as a valid low pulse and converts it to a 16 sclk cycles width of low pulse. Select an appropriate prescaler output clock that can detect a minimum 1.41 µs width of an input pulse.

Controlling serial data transfer

The control method to transmit/receive data in IrDA mode is the same as that of the normal interface. See previous sections for details on how to set and control the data formats, data transfers, and interrupts.

18.9 Details of Control Registers

Table 18.9.1 List of UART Registers

Address	Register name		Function
0x4100	UART_ST	UART Status Register	Indicates transfer, buffer and error statuses.
0x4101	UART_TXD	UART Transmit Data Register	Transmit data
0x4102	UART_RXD	UART Receive Data Register	Receive data
0x4103	UART_MOD	UART Mode Register	Sets transfer data format.
0x4104	UART_CTL	UART Control Register	Controls data transfer.
0x4105	UART_EXP	UART Expansion Register	Sets IrDA mode.

The following describes each UART register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4100: UART Status Register (UART_ST)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Status Register (UART_ST)	**0x4100** (8 bits)	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**FER**	Framing error flag	1	Error	0	Normal	0	R/W	Reset by writing 1.
		D5	**PER**	Parity error flag	1	Error	0	Normal	0	R/W	
		D4	**OER**	Overrun error flag	1	Error	0	Normal	0	R/W	
		D3	**RD2B**	Second byte receive flag	1	Ready	0	Empty	0	R	
		D2	**TRBS**	Transmit busy flag	1	Busy	0	Idle	0	R	Shift register status
		D1	**RDRY**	Receive data ready flag	1	Ready	0	Empty	0	R	
		D0	**TDBE**	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	

D7 Reserved

D6 FER: Framing Error Flag

Indicates whether a framing error has occurred or not.

1 (R): An error has occurred
0 (R): No error has occurred (default)
1 (W): Reset to 0
0 (W): Has no effect

When a framing error has occurred, FER is set to 1. A framing error occurs when data with a stop bit = 0 is received.

FER is reset by writing 1 or when RXEN (D0/UART_CTL register) is set to 0.

D5 PER: Parity Error Flag

Indicates whether a parity error has occurred or not.

1 (R): An error has occurred
0 (R): No error has occurred (default)
1 (W): Reset to 0
0 (W): Has no effect

When a parity error has occurred, PER is set to 1. The parity check function is effective only when PREN (D3/UART_MOD register) is set to 1. This check is performed when the received data is transferred from the shift register to the receive data buffer.

PER is reset by writing 1 or when RXEN (D0/UART_CTL register) is set to 0.

D4 OER: Overrun Error Flag

Indicates whether an overrun error has occurred or not.

1 (R): An error has occurred
0 (R): No error has occurred (default)
1 (W): Reset to 0
0 (W): Has no effect

When an overrun error has occurred, OER is set to 1. An overrun error will occur if new data is received when the receive data buffer is full and also if the shift register contains received data. When this error occurs, the shift register is overwritten with the new received data. The receive data in the buffer is left unchanged.

OER is reset by writing 1 or when RXEN (D0/UART_CTL register) is set to 0.

D3 RD2B: Second Byte Received Flag

Indicates that the receive data buffer contains two received data.

1 (R): Second byte is ready to read out
0 (R): Second entry is empty (default)

RD2B is set to 1 when the second data is loaded to the receive data buffer, and is reset to 0 when the first data is read out from the receive data buffer.

D2 TRBS: Transmit Busy Flag

Indicates the transmit shift register status.

1 (R): Busy
0 (R): Idle (default)

TRBS goes 1 when transmit data is loaded to the shift register from the transmit data buffer and returns to 0 upon completion of a data transmission. Read this flag to check whether the transmitter is busy or idle.

D1 RDRY: Receive Data Ready Flag

Indicates that the receive data buffer contains valid received data.

1 (R): Data is ready to read out
0 (R): Buffer is empty (default)

RDRY is set to 1 when received data is loaded to the receive data buffer, and is reset to 0 when all data are read out from the receive data buffer.

D0 TDBE: Transmit Data Buffer Empty Flag

Indicates the status of the transmit data buffer.

1 (R): Empty (default)
0 (R): Not empty

TDBE is reset to 0 when transmit data is written to the transmit data buffer and set to 1 when the transmit data in the buffer is transferred to the shift register.

0x4101: UART Transmit Data Register (UART_TXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Transmit Data Register (UART_TXD)	**0x4101** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	

D[7:0] TXD[7:0]: Transmit Data

Write transmit data to be set to the transmit data buffer. (Default: 0x0)

When data is written to this register, the UART starts transmitting. The data written to TXD[7:0] enters the transmit data buffer and waits for transmission. When the data in the transmit data buffer is transferred, a cause of transmit buffer empty interrupt occurs.

In 7-bit mode, TXD7 (MSB) is ignored.

The serial-converted data is output from the SOUT pin beginning with the LSB, in which the bits set to 1 are output as high-level signals and those set to 0 output as low-level signals.

This register can be read as well as written.

0x4102: UART Receive Data Register (UART_RXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Receive Data Register (UART_RXD)	**0x4102** (8 bits)	D7–0	**RXD[7:0]**	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.

D[7:0] RXD[7:0]: Receive Data

The data in the receive data buffer can be read from this register beginning with the older data first. The received data enters the receive data buffer. The receive data buffer is a two-byte FIFO and can receive data until it becomes full. When the buffer is full and also if the shift register contains received data, an overrun error will occur if the received data is not read by the time the next data receiving begins. The receive data buffer status flags RDRY (D1/UART_ST register) and RD2B (D3/UART_ST register) are provided to indicate that the receive data buffer contains valid received data and the second data, respectively.

When the receive data buffer has received the number of data specified with RBFI (D1/UART_CTL register), a cause of receive buffer full interrupt occurs.

In 7-bit mode, 0 is stored in RXD7.

The serial data input from the SIN pin is converted into parallel data beginning with the LSB, with the high-level signals changed to 1s and the low-level signals changed to 0s. The resulting data is stored in the receive data buffer.

This register is a read-only register, so no data can be written to it. (Default: 0x0)

0x4103: UART Mode Register (UART_MOD)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Mode Register (UART_MOD)	**0x4103** (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**CHLN**	Character length	1	8 bits	0	7 bits	0	R/W	
		D3	**PREN**	Parity enable	1	With parity	0	No parity	0	R/W	
		D2	**PMD**	Parity mode select	1	Odd	0	Even	0	R/W	
		D1	**STPB**	Stop bit select	1	2 bits	0	1 bit	0	R/W	
		D0	**SSCK**	Input clock select	1	External	0	Internal	0	R/W	

D[7:5] Reserved

D4 CHLN: Character Length Select Bit

Selects the character length of serial transfer data.

1 (R/W): 8 bits

0 (R/W): 7 bits (default)

D3 PREN: Parity Enable Bit

Enables the parity function.

1 (R/W): With parity

0 (R/W): No parity (default)

PREN is used to select whether the parity check for receive data will be performed or not, and whether a parity bit will be added to transmit data. When PREN is set to 1, the received data is checked for parity. A parity bit is automatically added to the transmit data. When PREN is set to 0, parity is not checked and no parity bit is added.

D2 PMD: Parity Mode Select Bit

Selects the parity mode.

1 (R/W): Odd parity

0 (R/W): Even parity (default)

Odd parity is selected by writing 1 to PMD, and even parity is selected by writing 0. Parity check and the addition of a parity bit are effective only when PREN (D3) is set to 1. If PREN (D3) = 0, settings of PMD do not have any effect.

D1 STPB: Stop Bit Select Bit

Selects a stop bit length.

1 (R/W): 2 bits

0 (R/W): 1 bit (default)

Two stop bits are selected by writing 1 to STPB, and one stop bit is selected by writing 0. The start bit is fixed at 1 bit.

D0 SSCK: Input Clock Select Bit

Selects the clock source.

1 (R/W): External clock (SCLK pin)

0 (R/W): Internal clock (default)

This bit is used to select the clock source between the internal clock (8-bit timer output clock) and an external clock (input from the SCLK pin). An external clock is selected by writing 1 to this bit, and an internal clock is selected by writing 0.

0x4104: UART Control Register (UART_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Control Register (UART_CTL)	**0x4104** (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		D6	**REIEN**	Receive error int. enable	1	Enable	0	Disable	0	R/W	
		D5	**RIEN**	Receive buffer full int. enable	1	Enable	0	Disable	0	R/W	
		D4	**TIEN**	Transmit buffer empty int. enable	1	Enable	0	Disable	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	**RBFI**	Receive buffer full int. condition	1	2 bytes	0	1 byte	0	R/W	
		D0	**RXEN**	UART enable	1	Enable	0	Disable	0	R/W	

D7 Reserved

D6 REIEN: Receive Error Interrupt Enable Bit

Enables an interrupt request to be output to the ITC when a receive error has occurred.

1 (R/W): Enable
0 (R/W): Disable (default)

Set this bit to 1 when processing receive errors in the interrupt handler routine.

D5 RIEN: Receive Buffer Full Interrupt Enable Bit

Enables an interrupt request to be output to the ITC when the receive data buffer receives the number of data specified by RBFI (D1).

1 (R/W): Enable
0 (R/W): Disable (default)

Set this bit to 1 when reading the received data in the interrupt handler routine.

D4 TIEN: Transmit Buffer Empty Interrupt Enable Bit

Enables an interrupt request to be output to the ITC when the transmit data written to the transmit data buffer is transferred to the shift register (when data transmission starts).

1 (R/W): Enable
0 (R/W): Disable (default)

Set this bit to 1 when writing transmit data to the transmit data buffer in the interrupt handler routine.

D[3:2] Reserved

D1 RBFI: Receive Buffer Full Interrupt Condition Setup Bit

Sets the number of data in the receive data buffer to generate a receive-buffer full interrupt.

1 (R/W): 2 bytes
0 (R/W): 1 byte (default)

When the specified number of received data is loaded to the receive data buffer, the UART outputs an interrupt request pulse to the ITC if the receive buffer full interrupt has been enabled (RIEN = 1). If RBFI is 0, an interrupt request pulse is output when a received data is loaded to the receive data buffer (when the RDRY flag (D1/UART_ST register) goes 1). If RBFI is 1, an interrupt request pulse is output when two received data occupy the receive data buffer (when the RD2B flag (D3/UART_ST register) goes 1).

D0 RXEN: UART Enable Bit

Enables the UART to transmit/receive data.

1 (R/W): Enable
0 (R/W): Disable (default)

Before the UART can transmit/receive data, RXEN must be set to 1. When RXEN is set to 0, data transmission/reception is disabled.
Always make sure RXEN = 0 before setting the transfer conditions.
Writing 0 to RXEN also clears the transmit/receive data buffers.

0x4105: UART Expansion Register (UART_EXP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
UART Expansion Register (UART_EXP)	**0x4105** (8 bits)	D7	–	reserved	–		–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock select	IRCLK[2:0] 0x7 0x6 0x5 0x4 0x3 0x2 0x1 0x0	Clock PCLK•1/128 PCLK•1/64 PCLK•1/32 PCLK•1/16 PCLK•1/8 PCLK•1/4 PCLK•1/2 PCLK•1/1	0x0	R/W	
		D3–1	–	reserved	–		–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1 On	0 Off	0	R/W	

D7 Reserved

D[6:4] IRCLK[2:0]: IrDA Receive Detection Clock Select Bits

These bits select a prescaler output clock as the input pulse detection clock.

Table 18.9.2 Selecting the IrDA Receive Detection Clock

IRCLK[2:0]	Prescaler output clock
0x7	PCLK•1/128
0x6	PCLK•1/64
0x5	PCLK•1/32
0x4	PCLK•1/16
0x3	PCLK•1/8
0x2	PCLK•1/4
0x1	PCLK•1/2
0x0	PCLK•1/1

(Default: 0x0)

This clock must be faster than the transfer clock sclk supplied from the 8-bit timer or input from the SCLK pin.

The demodulator regards a low pulse of which the width is longer than two cycles of the IrDA receive detection clock as a valid low pulse. Select an appropriate prescaler output clock that can detect a minimum 1.41 µs width of an input pulse.

D[3:1] Reserved

D0 IRMD: IrDA Mode Select Bit

Turns the IrDA interface function on and off.

1 (R/W): On

0 (R/W): Off (default)

Set this bit to 1 when using the IrDA interface. When set to 0, the module functions as a standard UART without IrDA.

18.10 Precautions

- Before setting the bits listed below, make sure the transmit and receive operations are disabled (RXEN = 0).
  - All bits (SSCK, STPB, PMD, PREN, and CHLN) of the UART_MOD register (0x4103)
  - All bits (RBFI, TIEN, RIEN, and REIEN except RXEN) of the UART_CTL register (0x4104)
  - All bits (IRMD and IRCLK[2:0]) of the UART_EXP register (0x4105)

  * **RXEN**: UART Enable Bit in the UART Control (UART_CTL) Register (D0/0x4104)
- When the UART is transmitting or receiving data, do not set RXEN to 0.
- The maximum transfer rate of the UART is limited to 115200 bps. Do not set a transfer rate that exceeds the limit.
- When the RXEN bit is set to 0 to disable transmit/receive operations, the transmit/receive data buffers are cleared (initialized). Therefore, make sure that the buffers do not contain any data waiting for transmission or reading before writing 0 to the RXEN bit.
- The IrDA receive detection clock must be faster than the transfer clock sclk supplied from the 8-bit timer or input from the SCLK pin.
- The demodulator regards a low pulse of which the width is longer than two cycles of the IrDA receive detection clock as a valid low pulse. Select an appropriate prescaler output clock as the IrDA receive detection clock so that it will be able to detect a minimum 1.41 µs width of an input pulse.

THIS PAGE IS BLANK.

19 SPI

19.1 Configuration of the SPI

The S1C17701 equipped with a synchronous serial interface module (hereafter SPI). The SPI module supports both master and slave modes and performs 8-bit serial data transfer. Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.

The SPI includes a transmit data buffer and a receive data buffer separately from the shift registers, and can generate two types of interrupts (transmit data buffer empty and receive data buffer full), this makes it possible to process continuous serial data transfers simply in an interrupt handler.

Figure 19.1.1 shows the structure of the SPI module.

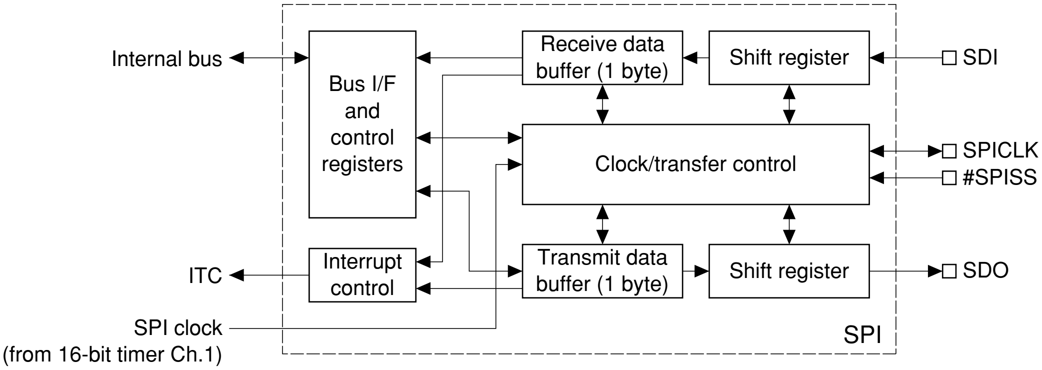

Internal bus
Bus I/F and control registers
Receive data buffer (1 byte)
Shift register
SDI
Clock/transfer control
SPICLK
#SPISS
ITC
Interrupt control
Transmit data buffer (1 byte)
Shift register
SDO
SPI clock (from 16-bit timer Ch.1)
SPI

Figure 19.1.1 Structure of SPI Module

19.2 SPI I/O Pins

Table 19.2.1 lists the SPI pins.

Table 19.2.1 List of SPI Pins

Pin name	I/O	Size	Function
SDI (P20)	I	1	SPI data input pin This pin inputs serial data from the SPI bus.
SDO (P21)	O	1	SPI data output pin This pin outputs serial data to the SPI bus.
SPICLK (P22)	I/O	1	SPI external clock input/output pin This pin outputs the SPI clock when the SPI is in master mode. This pin inputs an external clock when the SPI is in slave mode.
#SPISS (P17)	I	1	SPI slave select signal (active low) input pin A low level input to this pin selects this SPI device in slave mode.

The SPI input/output pins (SDI, SDO, SPICLK, #SPISS) are shared with the I/O ports (P20, P21, P22, P17) and they are initialized as general-purpose I/O port pins by default. Before using these pins for the SPI, the pin functions must be switched using the P2_PMUX and P1_PMUX registers. Set the control bits shown below to 1 to configure the pins for the SPI.

P20 → SDI

∗ **P20MUX**: P20 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D0/0x52a2)

P21 → SDO

∗ **P21MUX**: P21 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D1/0x52a2)

P22 → SPICLK

∗ **P22MUX**: P22 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D2/0x52a2)

P17 → #SPISS

∗ **P17MUX**: P17 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D7/0x52a1)

For details on switching pin function, see Section 10.2, “Selecting I/O Pin Functions (Port MUX).”

19.3 SPI Clock

In master mode, the SPI module uses the internal clock output from the 16-bit timer Ch.1 as the SPI clock. This clock drives the shift register and is output from the SPICLK pin to the slave device.
Program the 16-bit timer Ch.1 so that it will output a clock according to the transfer rate. See Chapter 11 "16-bit Timers (T16)," for controlling the timer.

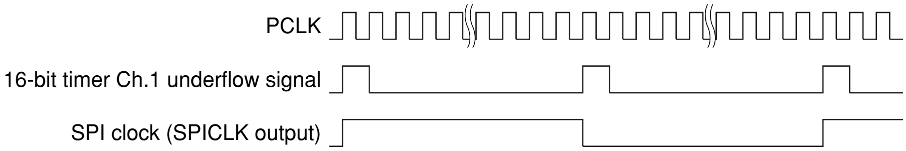


Figure 19.3.1 SPI Clock in Master Mode

In slave mode, the SPI module inputs the SPI clock from the SPICLK pin. Since the internal circuit operates with the internal PCLK clock, the input clock is differentiated and used to sync with the PCLK clock.

Note: The frequency of the clock input from the SPICLK pin must be 1/3 of PCLK or lower and the clock duty ratio must be 50%.

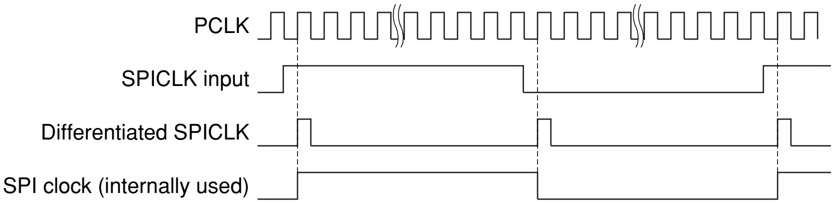


Figure 19.3.2 SPI Clock in Slave Mode

19.4 Setting the Data Transfer Conditions

The SPI module can be set in master or slave mode and the SPI clock polarity and phase can be set using the SPI_CTL register.
The data length is fixed at eight bits.

Note: Make sure that the SPI module is disabled (SPEN/SPI_CTL register = 0) before selecting master/slave mode and setting the clock conditions.

∗ **SPEN**: SPI Enable Bit in the SPI Control (SPI_CTL) Register (D0/0x4326)

Selecting master/slave mode

Use MSSL (D1/SPI_CTL register) to select whether the SPI module is set in master mode or slave mode. Setting MSSL to 1 selects master mode, and setting to 0 (default) selects slave mode. In master mode, the SPI performs data transfer using the internal clock. In slave mode, the SPI performs data transfer using a clock input from the master device.

∗ **MSSL**: Master/Slave Mode Select Bit in the SPI Control (SPI_CTL) Register (D1/0x4326)

Setting the SPI clock polarity and phase

Use CPOL (D2/SPI_CTL register) to select the SPI clock polarity. The SPI clock is configured as active low when CPOL is set to 1 or active high when CPOL is set to 0 (default).

∗ **CPOL**: Clock Polarity Select Bit in the SPI Control (SPI_CTL) Register (D2/0x4326)

The SPI clock phase is selected with CPHA (D3/SPI_CTL register).

∗ **CPHA**: Clock Phase Select Bit in the SPI Control (SPI_CTL) Register (D3/0x4326)

Setting these control bits determines the transfer timing as in the figure shown below.

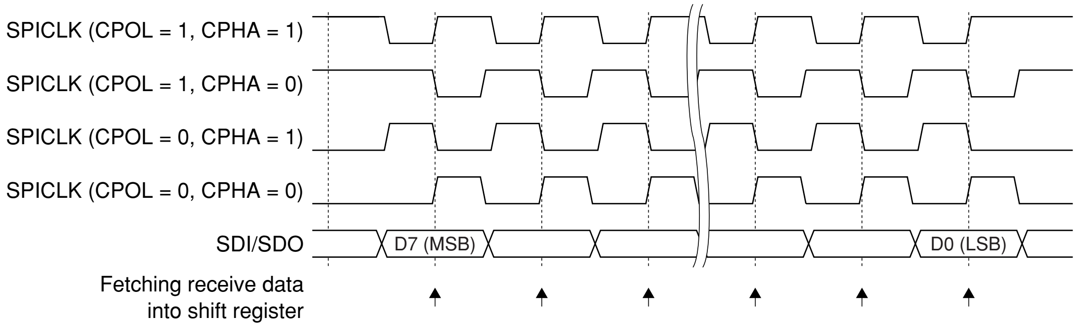

SPICLK (CPOL = 1, CPHA = 1)
SPICLK (CPOL = 1, CPHA = 0)
SPICLK (CPOL = 0, CPHA = 1)
SPICLK (CPOL = 0, CPHA = 0)
SDI/SDO
D7 (MSB)
D0 (LSB)
Fetching receive data into shift register

Figure 19.4.1 Clock and Data Transfer Timing

19.5 Data Transmit/Receive Control

Before starting data transfer, set up the conditions as shown below.

(1) Set up the 16-bit timer Ch.1 to output the SPI clock. See Chapter 11.
(2) Select either master or slave mode. See Section 19.4.
(3) Set up the clock conditions. See Section 19.4.
(4) Set up the interrupt conditions if the SPI interrupt is used. See Section 19.6.

Note: Make sure that the SPI module is disabled (SPEN/SPI_CTL register = 0) before setting the conditions above.

∗ **SPEN**: SPI Enable Bit in the SPI Control (SPI_CTL) Register (D0/0x4326)

Enabling data transmission/reception

First, set the SPEN bit (D0/SPI_CTL register) to 1 to enable SPI operation. This puts the SPI in ready-to-transmit/receive status and enables clock input/output.

Note: Do not set the SPEN bit to 0 while the SPI module is transmitting/receiving data.

Data transmit control

To start transmission, write transmit data to the SPI_TXD register (0x4322).

∗ **SPI_TXD**: SPI Transmit Data Register (0x4322)

Data is written to the transmit data buffer and the SPI starts data transmission.
The buffered data is sent to the shift register for transmission. In master mode, the SPI module starts outputting the clock from the SPICLK pin. In slave mode, the SPI module waits for clock input from the SPICLK pin. The data bits in the shift register are shifted one by one at the rising or falling edge of the clock configured with CPHA (D3/SPI_CTL register) and CPOL (D2/SPI_CTL register) (see Figure 19.4.1), and are output from the SDO pin. The MSB of data is transmitted first.

∗ **CPHA**: Clock Phase Select Bit in the SPI Control (SPI_CTL) Register (D3/0x4326)
∗ **CPOL**: Clock Polarity Select Bit in the SPI Control (SPI_CTL) Register (D2/0x4326)

The SPI module provides two status flags for data transmit control, SPTBE (D0/SPI_ST register) and SPBSY (D2/SPI_ST register).

∗ **SPTBE**: Transmit Data Buffer Empty Flag in the SPI Status (SPI_ST) Register (D0/0x4320)
∗ **SPBSY**: Transfer Busy Flag in the SPI Status (SPI_ST) Register (D2/0x4320)

The SPTBE flag indicates the transmit data buffer status; it goes 0 when the application program writes data to the SPI_TXD register (transmit data buffer) and returns to 1 when the data in the transmit data buffer is sent to the shift register for transmitting. An interrupt can be generated when this flag goes 1 (see Section 19.6). Use this interrupt or read the SPTBE flag to check that the transmit data buffer becomes empty when transmitting the next data. Although the transmit data buffer size is one byte, transmit data can be written while the previous data is being transmitted as the shift register is separately provided. However, make sure that the transmit data buffer is empty before writing transmit data. If data is written when the SPTBE flag is 0, the previous transmit data in the transmit data buffer is overwritten with the new data.

In master mode, the SPBSY flag indicates the shift register status; it goes 1 when transmit data is loaded from the transmit data buffer and returns to 0 upon completion of data transmission. Read this flag to check whether the SPI module is busy or idle.
In slave mode, the SPBSY flag indicates the SPI slave select signal (#SPISS pin) status; it goes 1 when this SPI module is selected as a slave or goes 0 when this SPI module is deselected.

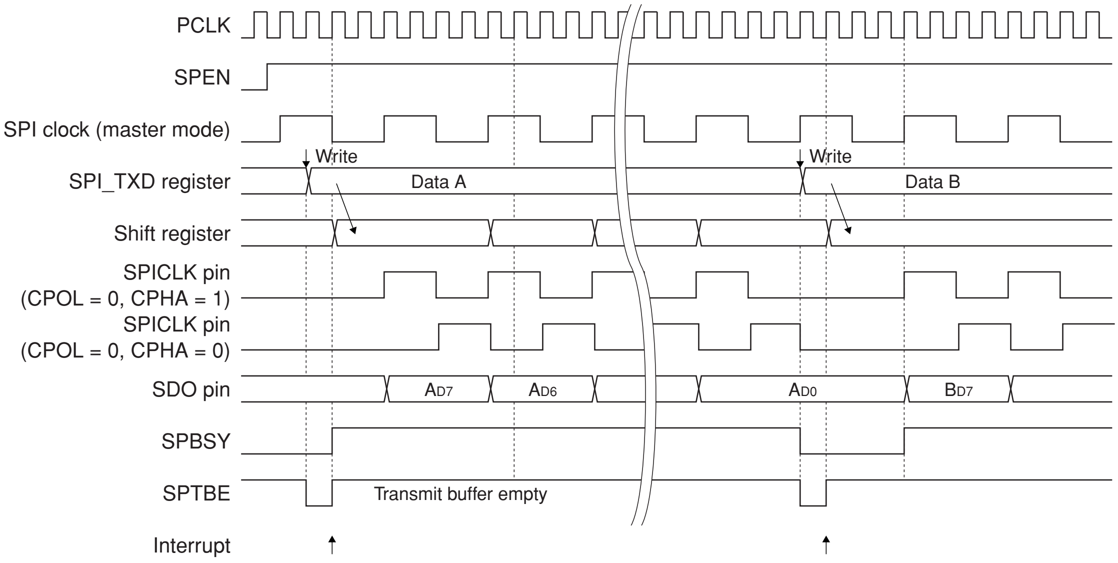


Figure 19.5.1 Data Transmit Timing Chart

Data receive control

In master mode, write dummy data to the SPI_TXD register (0x4322). Writing to the SPI_TXD register is used as the trigger for data receiving as well as starting data transmission. Also actual data to be transmitted can be written as the SPI module performs data transmission and reception simultaneously.
The SPI module starts output of the SPI clock from the SPICLK pin.

In slave mode, the SPI module waits for clock input from the SPICLK pin. When receiving data in slave mode without any data transmission, it is not necessary to write data to the SPI_TXD register. The receive process activates by the clock input from the master device. When performing data transmission and reception simultaneously, the transmit data should be written to the SPI_TXD register before a clock is input.

The data bits are fetched in the shift register one by one at the rising or falling edge of the clock configured with CPHA (D3/SPI_CTL register) and CPOL (D2/SPI_CTL register) (see Figure 19.4.1). The MSB of data is received first.
When eight data bits are received in the shift register, the received data is loaded into the receive data buffer.

The received data in the buffer can be read from the SPI_RXD register (0x4324).

∗ **SPI_RXD**: SPI Receive Data Register (0x4324)

The SPI module provides the SPRBF flag (D1/SPI_ST register) for data receive control.

∗ **SPRBF**: Receive Data Buffer Full Flag in the SPI Status (SPI_ST) Register (D1/0x4320)

The SPRBF flag indicates the receive data buffer status; it goes 1 when the data received in the shift register is loaded to the receive data buffer to indicate that the receive data can be read and returns to 0 when the data in the receive data buffer is read out from the SPI_RXD register. An interrupt can be generated when this flag goes 1 (see Section 19.6). Use this interrupt or read the SPRBF flag to check that the receive data buffer contains valid data when reading received data. Although the receive data buffer size is one byte, the previous received data can be maintained while the next data is being received as the shift register is separately provided. However, be sure to read the receive data before the next data has been received. If the next data is received before the previous received data in the receive data buffer has been read, the previous received data is overwritten with the new data.

In master mode, the SPBSY flag that indicates the shift register status can be used as in data transmission.

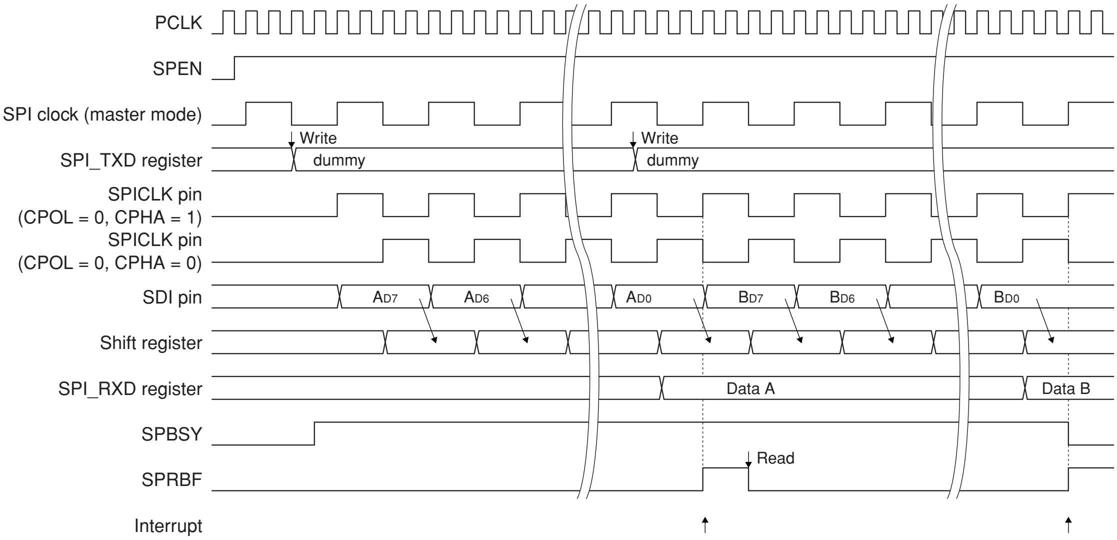


Figure 19.5.2 Data Receive Timing Chart

Disabling data transmission/reception

After data transfer (both transmission and reception) has finished, write 0 to the SPEN bit to disable data transmission/reception.

Always make sure that the SPTBE flag is 1 and SPRBF flag is 0 before data transmission/reception is disabled.

When the SPEN bit is set to 0, the transmit and receive data buffers are placed in empty status (data is cleared if any remains). Furthermore, the data being transferred cannot be guaranteed if SPEN is set to 0 during transmitting/receiving.

19.6 SPI Interrupt

The SPI module can generate the following two types of interrupts:
- Transmit buffer empty interrupt
- Receive buffer full interrupt

The SPI module has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with the two causes of interrupt. To determine the cause of interrupt that has occurred, read the status flags.

Transmit buffer empty interrupt

Set the SPTIE bit (D4/SPI_CTL register) to 1 when using this interrupt. If SPTIE is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **SPTIE**: Transmit Data Buffer Empty Interrupt Enable Bit in the SPI Control (SPI_CTL) Register (D4/0x4326)

When the transmit data set in the transmit data buffer is transferred to the shift register, the SPI module sets the SPTBE bit (D0/SPI_ST register) to 1 to indicate that the transmit data buffer is empty. At the same time, the SPI module outputs an interrupt request pulse to the ITC if the transmit buffer empty interrupt has been enabled (SPTIE = 1).

∗ **SPTBE**: Transmit Data Buffer Empty Flag in the SPI Status (SPI_ST) Register (D0/0x4320)

If other interrupt conditions are satisfied, an interrupt is generated.
The SPI interrupt handler routine should read the SPTBE flag to check if the interrupt has occurred due to a transmit buffer empty or another cause. When SPTBE = 0, the SPI interrupt handler routine can write the next transmit data to the transmit data buffer.

Receive buffer full interrupt

Set the SPRIE bit (D5/SPI_CTL register) to 1 when using this interrupt. If SPRIE is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **SPRIE**: Receive Data Buffer Full Interrupt Enable Bit in the SPI Control (SPI_CTL) Register (D5/0x4326)

When data received in the shift register is loaded to the receive data buffer, the SPI module sets the SPRBF bit (D1/SPI_ST register) to 1 to indicate that the received data buffer is full. At the same time, the SPI module outputs an interrupt request pulse to the ITC if the receive buffer full interrupt has been enabled (SPRIE = 1).

∗ **SPRBF**: Receive Data Buffer Full Flag in the SPI Status (SPI_ST) Register (D1/0x4320)

If other interrupt conditions are satisfied, an interrupt is generated.
The SPI interrupt handler routine should read the SPRBF flag to check if the interrupt has occurred due to a receive buffer full or another cause. When SPRBF = 1, the SPI interrupt handler routine can read the received data from the receive data buffer.

ITC registers for SPI interrupts

The following shows the control bits of the ITC provided for the SPI module:

Interrupt flag

∗ **IIFT6**: SPI Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D14/0x4300)

Interrupt enable bit

∗ **IIEN6**: SPI Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D14/0x4302)

Interrupt level setup bits

∗ **IILV6[2:0]**: SPI Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV3) Register 3 (D[2:0]/0x4314)

When the SPI outputs an interrupt request pulse, the interrupt flag IIFT6 is set to 1.
If the interrupt enable bit IIEN6 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the SPI interrupt, set the IIEN6 bit to 0.
The IIFT6 flag is always set to 1 by the SPI interrupt request pulse, regardless of how the IIEN6 bit is set (even when set to 0).
The interrupt level setup bits IILV6[2:0] set the interrupt level (0 to 7) of the SPI interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The SPI interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Chapter 6, “Interrupt Controller (ITC).”

Interrupt vector

The following shows the vector number and vector address for the SPI interrupt:

Vector number: 18 (0x12)
Vector address: 0x8048

19.7 Details of Control Registers

Table 19.7.1 List of SPI Registers

Address	Register name		Function
0x4320	SPI_ST	SPI Status Register	Indicates transfer and buffer statuses.
0x4322	SPI_TXD	SPI Transmit Data Register	Transmit data
0x4324	SPI_RXD	SPI Receive Data Register	Receive data
0x4326	SPI_CTL	SPI Control Register	Sets the SPI mode and enables data transfer.

The following describes each SPI register. These are all 16-bit registers.

Notes:
- When setting the registers, be sure to write a 0, and not a 1, for all "reserved bits."
- Be sure to use 16-bit access instructions for reading/writing from/to the SPI registers. The SPI registers do not allow reading/writing using 32-bit and 8-bit access instructions.

0x4320: SPI Status Register (SPI_ST)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Status Register (SPI_ST)	**0x4320** (16 bits)	D15–3	–	reserved	–				–	–	0 when being read.
		D2	**SPBSY**	Transfer busy flag (master)	1	Busy	0	Idle	0	R	
				ss signal low flag (slave)	1	ss = L	0	ss = H			
		D1	**SPRBF**	Receive data buffer full flag	1	Full	0	Not full	0	R	
		D0	**SPTBE**	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	

D[15:3] Reserved

D2 SPBSY: Transfer Busy Flag (Master Mode)/ss Signal Low Flag (Slave Mode)

Master mode

Indicates the SPI transmit/receive operation status.

1 (R): Busy

0 (R): Idle (default)

SPBSY is set to 1 when the SPI starts data transmission/reception in master mode and stays 1 while data transmission/reception is in progress. SPBSY is reset to 0 upon completion of the transmit/receive operation.

Slave mode

Indicates the slave select (#SPISS) signal status.

1 (R): Low level (this SPI is selected)

0 (R): High level (this SPI is deselected) (default)

SPBSY is set to 1 when the master device activates the #SPISS signal to select this SPI module (slave device), and is reset to 0 when the master device negates the #SPISS signal to deselect this SPI module.

D1 SPRBF: Receive Data Buffer Full Flag

Indicates the receive data buffer status.

1 (R): Full

0 (R): Not full (default)

SPRBF is set to 1 when the data received in the shift register is loaded to the receive data buffer (receive operation completed), indicating that the received data can be read out. This bit is reset to 0 when the data is read out from the SPI_RXD register (0x4324).

D0 SPTBE: Transmit Data Buffer Empty Flag

Indicates the transmit data buffer status.

1 (R): Empty (default)

0 (R): Not empty

SPTBE is reset to 0 when transmit data is written to the SPI_TXD register (transmit data buffer, 0x4322) and is set to 1 when the written data is transferred to the shift register (transmit operation started).

Transmit data should be written to the SPI_TXD register when this bit = 1.

0x4322: SPI Transmit Data Register (SPI_TXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Transmit Data Register (SPI_TXD)	**0x4322** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPTDB[7:0]**	SPI transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff	0x0	R/W	

D[15:8] Reserved

D[7:0] SPTDB[7:0]: SPI Transmit Data Buffer Bits

Set transmit data to be written to the transmit data buffer. (Default: 0x0)

In master mode, data transmission begins by writing data to this register. In slave mode, the register contents are transferred to the shift register to start data transmission when a clock is input from the master device.

SPTBE (D0/SPI_ST register) is set to 1 (empty) when the data is transferred to the shift register. At the same time, a cause of transmit data buffer empty interrupt occurs. The next transmit data can be written to the register at any time thereafter, even when the SPI is sending data.

The serial-converted data is output from the SDO pin beginning with the MSB, in which the bits set to 1 are output as high-level signals and those set to 0 output as low-level signals.

0x4324: SPI Receive Data Register (SPI_RXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Receive Data Register (SPI_RXD)	**0x4324** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPRDB[7:0]**	SPI receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff	0x0	R	

D[15:8] Reserved

D[7:0] SPRDB[7:0]: SPI Receive Data Buffer Bits

Stores received data. (Default: 0x0)

When a receive operation is completed and the data received in the shift register is loaded to the receive data buffer, SPRBF (D1/SPI_ST register) is set to 1 (buffer full). At the same time, a cause of receive data buffer full interrupt occurs. Thereafter, the data can be read out at any time before a receive operation for the next data is completed.

If the next data receive operation is completed before this register is read out, the data in it is overwritten with the newly received data.

The serial data input from the SDI pin is converted into parallel data beginning with the MSB, with the high-level signals changed to 1s and the low-level signals changed to 0s. The resulting data is stored in this register.

SPI_RXD is a read-only register, so no data can be written to it.

0x4326: SPI Control Register (SPI_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Control Register (SPI_CTL)	**0x4326** (16 bits)	D15–6	–	reserved	–				–	–	0 when being read.
		D5	**SPRIE**	Receive data buffer full int. enable	1	Enable	0	Disable	0	R/W	
		D4	**SPTIE**	Transmit data buffer empty int. enable	1	Enable	0	Disable	0	R/W	
		D3	**CPHA**	Clock phase select	1	Data out	0	Data in	0	R/W	These bits must be set before setting SPEN to 1.
		D2	**CPOL**	Clock polarity select	1	Active L	0	Active H	0	R/W	
		D1	**MSSL**	Master/slave mode select	1	Master	0	Slave	0	R/W	
		D0	**SPEN**	SPI enable	1	Enable	0	Disable	0	R/W	

D[15:6] Reserved

D5 SPRIE: Receive Data Buffer Full Interrupt Enable Bit

Enables/disables SPI interrupt caused by receive data buffer full.

1 (R/W): Enable

0 (R/W): Disable (default)

When SPRIE is set to 1, SPI (receive data buffer full) interrupt requests to the ITC are enabled. A receive data buffer full interrupt request occurs when the data received in the shift register is loaded to the receive data buffer (receive operation completed).

When SPRIE is set to 0, SPI interrupts caused by receive data full are not generated.

D4 SPTIE: Transmit Data Buffer Empty Interrupt Enable Bit

Enables/disables SPI interrupt caused by transmit data buffer empty.

1 (R/W): Enable

0 (R/W): Disable (default)

When SPTIE is set to 1, SPI (transmit data buffer empty) interrupt requests to the ITC are enabled. A transmit data buffer empty interrupt request occurs when the data written to the transmit data buffer is transferred to the shift register (transmit operation started).

When SPTIE is set to 0, SPI interrupts caused by transmit data buffer empty are not generated.

D3 CPHA: SPI Clock Phase Select Bit

Selects the phase of the SPI clock. (Default: 0)

This bit controls the data transfer timing in conjunction with the CPOL (D2) bit (see Figure 19.7.1).

D2 CPOL: SPI Clock Polarity Select Bit

Selects the polarity of the SPI clock.

1 (R/W): Active low

0 (R/W): Active high (default)

This bit controls the data transfer timing in conjunction with the CPHA (D3) bit (see Figure 19.7.1).

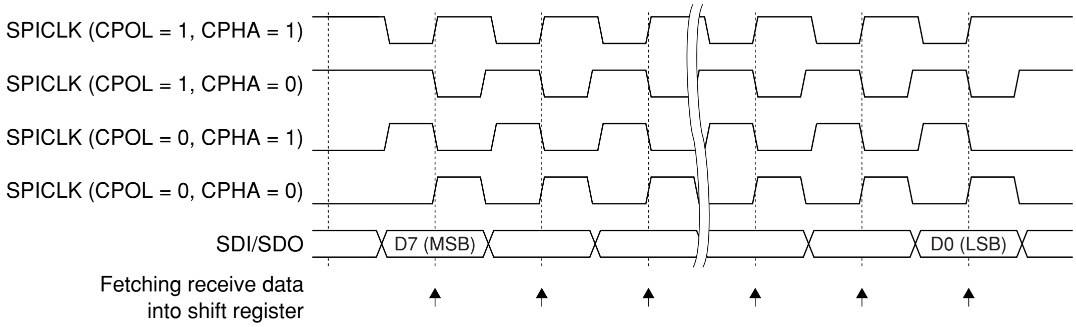

SPICLK (CPOL = 1, CPHA = 1)
SPICLK (CPOL = 1, CPHA = 0)
SPICLK (CPOL = 0, CPHA = 1)
SPICLK (CPOL = 0, CPHA = 0)
SDI/SDO
D7 (MSB)
D0 (LSB)
Fetching receive data into shift register

Figure 19.7.1 Clock and Data Transfer Timing

D1 MSSL: Master/Slave Mode Select Bit

Sets the SPI module in master or slave mode.
1 (R/W): Master mode
0 (R/W): Slave mode (default)

Setting MSSL to 1 selects master mode, and setting to 0 selects slave mode. In master mode, the SPI performs data transfer using the clock generated by the 16-bit timer Ch.1. In slave mode, the SPI performs data transfer using a clock input from the master device.

D0 SPEN: SPI Enable Bit

Enables/disables operation of the SPI module.
1 (R/W): Enable
0 (R/W): Disable (default)

When SPEN is set to 1, the SPI module is activated and data transfer is enabled.
When SPEN is set to 0, the SPI module goes off.

Note: Make sure that the SPEN bit is 0 before setting the CPHA, CPOL, and MSSL bits.

19.8 Precautions

- Be sure to use 16-bit access instructions for reading/writing from/to the SPI registers (0x4320 to 0x4326). The SPI registers do not allow reading/writing using 32-bit and 8-bit access instructions.
- Do not access the SPI_CTL register (0x4326), while the SPBSY flag (D2/SPI_ST register) is set to 1 (during data transfer).

  ∗ **SPBSY**: Transfer Busy Flag in the SPI Status (SPI_ST) Register (D2/0x4320)

20 I2C

20.1 Configuration of the I2C

The S1C17701 equipped with an I2C bus interface module for high-speed synchronous serial communication. This I2C module operates as a master using the clock supplied from the 16-bit timer Ch.2 (supports single master mode only). It supports standard (100 kbps) and fast (400 kbps) modes, and 7-bit/10-bit slave addressing. The I2C module includes a noise remove function to secure reliable data transfer.

Also it can generate two types of interrupts (transmit buffer empty and receive buffer full interrupts), this makes it possible to process continuous serial data transfer simply in an interrupt handler.

Figure 20.1.1 shows the structure of the I2C module.

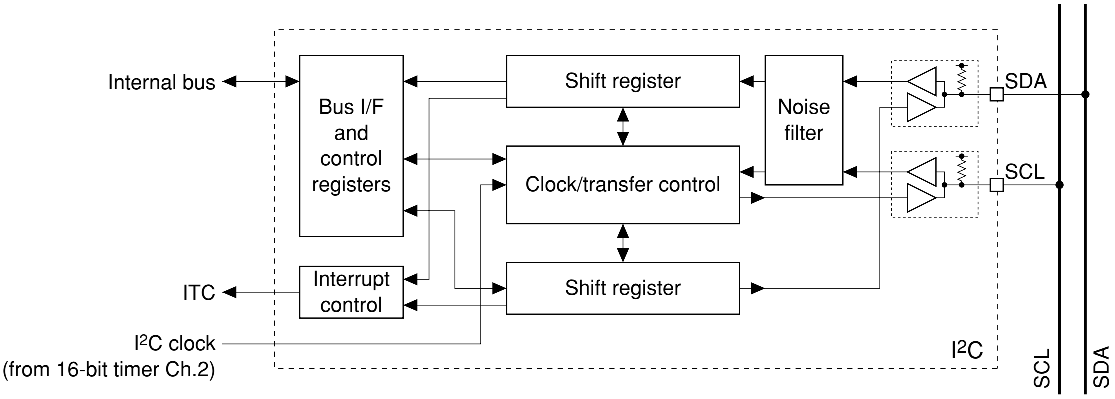


Figure 20.1.1 Structure of I2C Module

Note: The I2C module does not have a clock stretch function. Therefore, it does not support I2C slave devices that use clock stretch for synchronization of data communication.

20.2 I^{2}C I/O Pins

Table 20.2.1 lists the I^{2}C pins.

Table 20.2.1 List of I^{2}C Pins

Pin name	I/O	Size	Function
SDA (P14)	I/O	1	I^{2}C data input/output pin This pin inputs serial data from the I^{2}C bus and outputs serial data to the I^{2}C bus.
SCL (P15)	I/O	1	I^{2}C clock input/output pin This pin inputs the SCL line status and outputs the serial clock to the I^{2}C bus.

The I^{2}C input/output pins (SDA, SCL) are shared with the I/O ports (P14, P15) and they are initialized as general-purpose I/O port pins by default. Before using these pins for the I^{2}C, the pin functions must be switched using the P1_PMUX register. Set the control bits shown below to 1 to configure the pins for the I^{2}C.

P14 → SDA

∗ **P14MUX**: P14 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D4/0x52a1)

P15 → SCL

∗ **P15MUX**: P15 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D5/0x52a1)

For details on switching pin function, see Section 10.2, “Selecting I/O Pin Functions (Port MUX).”

20.3 I^{2}C Clock

The I^{2}C module uses the internal clock output from the 16-bit timer Ch.2 as the synchronous clock. This clock drives the shift register and is output from the SCL pin to the slave I^{2}C device.
Program the 16-bit timer Ch.2 so that it will output a clock according to the transfer rate. Refer to Chapter 11, "16-bit Timers (T16)," for controlling the 16-bit timer.

The I^{2}C module does not function as a slave device. The SCL input is used to check the SCL status of the I^{2}C bus but it is not used to input synchronous clock.

20.4 Setting before Starting Data Transfer

The I2C module has a noise remove function selectable in the application program.

Noise remove function

The I2C module contains a function to remove noise from the SDA and SCL input signals. This function is enabled by setting NSERM (D4/I2C_CTL register) to 1.

Note, however, that the I2C clock (16-bit timer Ch.2 output clock) frequency must be a 1/6 of PCLK or lower to use the noise remove function.

∗ **NSERM**: Noise Remove On/Off Bit in the I2C Control (I2C_CTL) Register (D4/0x4342)

20.5 Data Transmit/Receive Control

Before starting data transfer, set up the conditions by the procedure below.

(1) Set up the 16-bit timer Ch.2 to output the I^{2}C clock. See Chapter 11.

(2) Select optional functions. See Section 20.4.

(3) Set up the interrupt conditions if the I^{2}C interrupt is used. See Section 20.6.

Note: Make sure that the I^{2}C module is disabled (I2CEN/I2C_EN register = 0) before setting the conditions above.

∗ **I2CEN**: I^{2}C Enable Bit in the I^{2}C Enable (I2C_EN) Register (D0/0x4340)

Enabling data transmission/reception

First, set the I2CEN bit (D0/I2C_EN register) to 1 to enable I^{2}C operation. This makes the I^{2}C in ready-to-transmit/receive status and enables clock output.

Note: Do not set the I2CEN bit to 0 while the I^{2}C module is transmitting/receiving data.

Starting data transmission/reception

To start data transmission, the I^{2}C master (this module) must generate a START condition, and then send a slave address to establish the communication.

(1) Register setting procedure

To generate a START condition, set the following registers in the order shown below:

1. Set the slave address to RTDT[7:0] (D[7:0]/I2C_DAT register). (First transmit data when 10-bit address is used; see Figure 20.5.2.)
2. Set TXE (D9/I2C_DAT register) to 1.
3. Set STRT (D0/I2C_CTL register) to 1.

∗ **RTDT[7:0]**: Receive/Transmit Data Bits in the I^{2}C Data (I2C_DAT) Register (D[7:0]/0x4344)
∗ **TXE**: Transmit Execution Bit in the I^{2}C Data (I2C_DAT) Register (D9/0x4344)
∗ **STRT**: Start Control Bit in the I^{2}C Control (I2C_CTL) Register (D0/0x4342)

This procedure generates the communication waveforms as shown in Items (2) and (3) below. Be sure to follow the register setting procedure.

(2) Generating a START condition

The START condition is a state in which the SDA line is pulled down to low with the SCL line held at high.

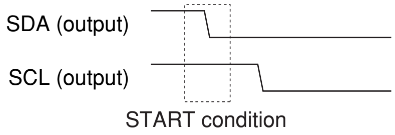


Figure 20.5.1 START Condition

Set the STRT bit (D0/I2C_CTL register) to 1 to generate a START condition.

∗ **STRT**: Start Control Bit in the I^{2}C Control (I2C_CTL) Register (D0/0x4342)

After a START condition has been generated, STRT is automatically reset to 0.

(3) Sending a slave address

After a START condition has been generated, the I^{2}C master (this module) sends the address of the slave to communicate and a bit to specify data transfer direction. The I^{2}C module supports two slave address sizes: 7-bit slave address and 10-bit slave address. The I^{2}C module sends the slave address bits with the transfer direction bit using the 8-bit transmit/receive data register. So one 7-bit slave address can be sent at a time. A 10-bit address should be sent in two parts with software control. Figure 20.5.2 shows the address data format.

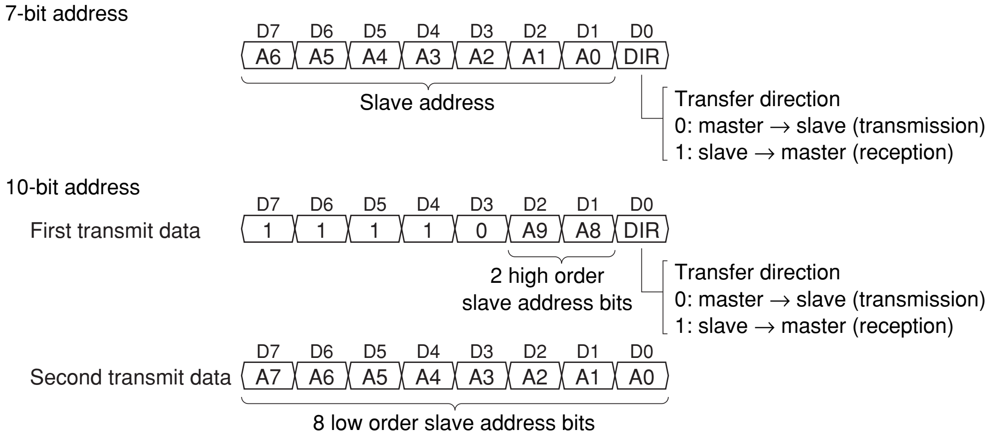


Figure 20.5.2 Transmit Data to Specify Slave Address and Data Direction

The transfer direction bit specifies the direction for the data transfer that follows the slave address transfer. Set the transfer direction bit to 0 when transmitting data from the master to the slave; set it to 1 when receiving data from the slave.
Configure an 8-bit data as above and set it into the transmit/receive data register. After that control data transmission as described below.

The slave address with a transfer direction bit must be sent once after a START condition has been generated.
After a slave address has been sent, perform data transmission or data reception as many times as necessary. It is necessary to perform data transmission or data reception according to the transfer direction specified with the slave address.

Data transmit control

The following explains how to transmit data. The slave address should be sent in the same way.
To transmit byte data, set the data to the RTDT[7:0] bits (D[7:0]/I2C_DAT register). At the same time, set the TXE bit (D9/I2C_DAT register) to 1 to execute one byte data transmission.

* **RTDT[7:0]**: Receive/Transmit Data Bits in the I2C Data (I2C_DAT) Register (D[7:0]/0x4344)
* **TXE**: Transmit Execution Bit in the I2C Data (I2C_DAT) Register (D9/0x4344)

When the TXE bit is set to 1, the I2C module starts data transmission in sync with the clock. If a START condition is being generated or the previous data is being transferred, the I2C module starts data transmission after waiting for completion of the process.
First, the I2C module transfer the written data to the shift register and starts outputting the clock from the SCL pin. At this time, TXE is reset to 0 and a cause of interrupt occurs. This allows the program to set the next transmit data and TXE again.
The data bits in the shift register are shifted one by one at the falling edge of the clock and are output from the SDA pin. The MSB is transmitted first.
The I2C module outputs nine clocks for one data transmission. In the ninth clock cycle, the I2C module sets the SDA signal into high-impedance status to input an ACK or NACK bit from the slave.
If the slave could receive byte data, it returns an ACK (0) bit to the master. If the slave could not receive byte data, the SDA line is not pulled down. The I2C module regards this status as a NACK (1) returned (transmission fails).

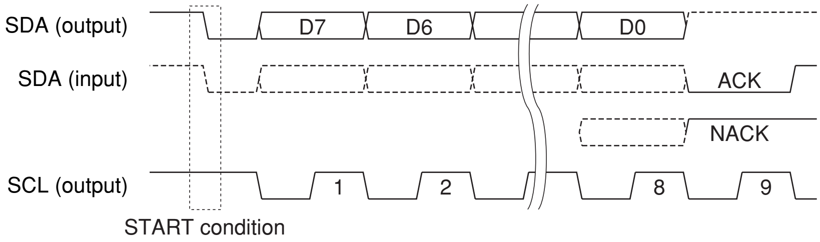


Figure 20.5.3 ACK and NACK

The I²C module provides two status bits for data transmit control, TBUSY flag (D8/I2C_CTL register) and RTACK bit (D8/I2C_DAT register).

- ∗ **TBUSY**: Transmit Busy Flag in the I²C Control (I2C_CTL) Register (D8/0x4342)
- ∗ **RTACK**: Receive/Transmit ACK Bit in the I²C Data (I2C_DAT) Register (D8/0x4344)

The TBUSY flag indicates the data transmit status; it goes 1 when data transmission (including slave address transmission) starts and returns to 0 upon completion of data transmission. Also TBUSY returns to 0 in wait state. Read this flag to check whether the I²C module is busy or idle.
The RTACK bit indicates whether the slave returned ACK or not in a previous data transmission; it goes 0 when an ACK bit is received or goes 1 if an ACK bit is not received.

Data receive control

The following explains how to receive data. Even in data reception, a START condition must be generated and a slave address with the transfer direction bit set to 1 must be sent before starting data reception.
To receive byte data, set the RXE bit (D10/I2C_DAT register) to 1 to execute receiving one byte data. RXE can be set to 1 at the same time when TXE (D9/I2C_DAT register) is set to 1 for sending a slave address. If TXE and RXE are both set to 1, TXE is effective.

- ∗ **RXE**: Receive Execution Bit in the I²C Data (I2C_DAT) Register (D10/0x4344)

When the RXE bit is set to 1 and the I²C module is ready to receive, the I²C module sets the SDA line into high-impedance and starts outputting the clock from the SCL pin. The data bits are fetched in the shift register one by one at the rising edge of the clock. The MSB is received first.
RXE is reset to 0 during fetching D6.
When eight data bits are received in the shift register, the received data is loaded into RTDT[7:0].
The I²C module provides two status bits for data receive control, RBRDY (D11/I2C_DAT register) and RBUSY (D9/I2C_CTL register).

- ∗ **RBRDY**: Receive Buffer Ready Bit in the I²C Data (I2C_DAT) Register (D11/0x4344)
- ∗ **RBUSY**: Receive Busy Flag in the I²C Control (I2C_CTL) Register (D9/0x4342)

The RBRDY flag indicates the received data status; it goes 1 when the received data in the shift register is loaded into RTDT[7:0] and returns to 0 when the received data is read from RTDT[7:0]. An interrupt can be generated when this flag goes 1. Use this interrupt or read the RBRDY flag to check that RTDT[7:0] contains valid data when reading received data. If the next data has been received before the previous received data in RTDT[7:0] is read, the previous received data is overwritten with the new data.
The RBUSY flag indicates the data receive operation status; it goes 1 when data reception starts and returns to 0 upon completion of data reception. Also RBUSY returns to 0 in wait state. Read this flag to check whether the I²C module is busy or idle.
The I²C module outputs nine clocks for one data transmission. In the ninth clock cycle, the I²C module sends an ACK or NACK bit from the SDA pin to the slave. The bit status to be sent can be set to RTACK (D8/I2C_DAT register). Set RTACK to 0 to send ACK or set to 1 to send NACK.

Terminating data transmission/reception

(1) Generating a STOP condition

To terminate data transfer after all data have been sent/received, the I²C master (this module) must generate a STOP condition. The STOP condition is a state in which the SDA line is pulled up from low to high with the SCL line held at high.

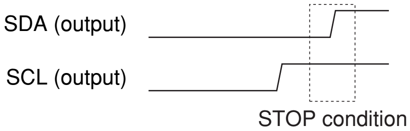


Figure 20.5.4 STOP Condition

Set the STP bit (D1/I2C_CTL register) to 1 to generate a STOP condition.

- ∗ **STP**: Stop Control Bit in the I²C Control (I2C_CTL) Register (D1/0x4342)

When STP is set to 1, the I²C module pulls up the SDA line from low to high with the SCL line held at high to generate a STOP condition on the I²C bus. This makes the I²C bus in free status.
Furthermore, the I²C module allows presetting for generating a STOP condition in advance. To do this, set STP to 1 after checking if the I²C is operating (TBUSY = 1 or RBUSY = 1). A STOP condition will be generated upon completion of data transmission/reception (including an ACK transfer).
STP is automatically reset to 0 after a STOP condition has been generated.

The I²C module does not support repeated START condition. A STOP condition cannot be omitted before generating a new START condition for starting the next data transfer.

Wait state by setting TXE, RXE, STRT, and STP

If TXE (D9/I2C_DAT register), RXE (D10/I2C_DAT register), STRT (D0/I2C_CTL register), and STP (D1/I2C_CTL register) have all been set to 0 when byte data and ACK transfer have finished, the I²C module fixes the SCL output at low and enters wait state. The wait state will be canceled by writing 1 to TXE or RXE to resume data transmission/reception or by writing 1 to STP to generate a STOP condition.

Disabling data transmission/reception

After data transfer (both transmission and reception) has finished, write 0 to the I2CEN bit to disable data transmission/reception.
Always make sure that the RBUSY and TBUSY flags are 0 before data transmission/reception is disabled.
The data being transferred cannot be guaranteed if I2CEN is set to 0 during transmitting/receiving.

Timing charts

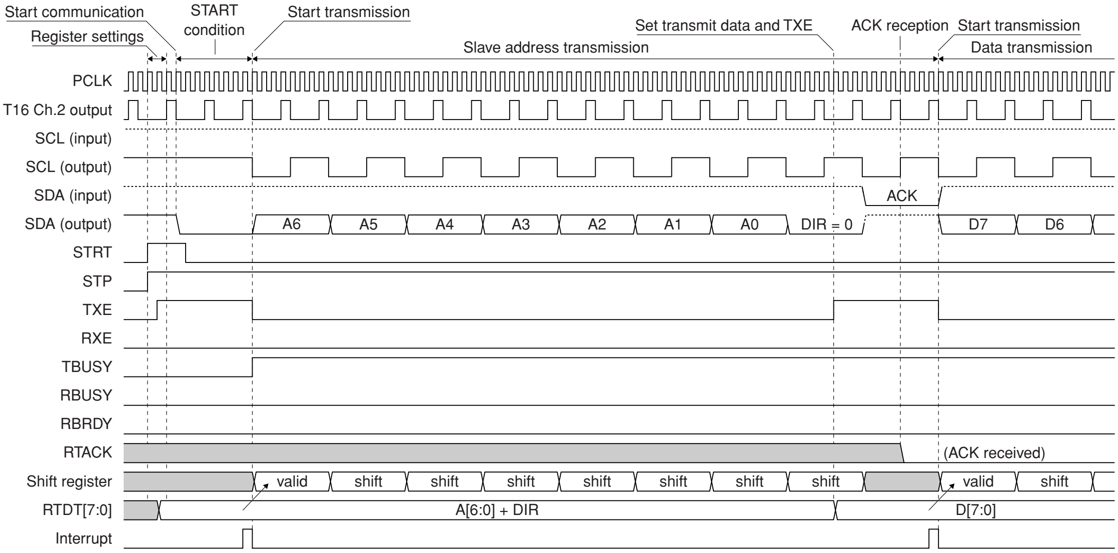


Figure 20.5.5 I²C Timing Chart 1 (START condition → data transmission)

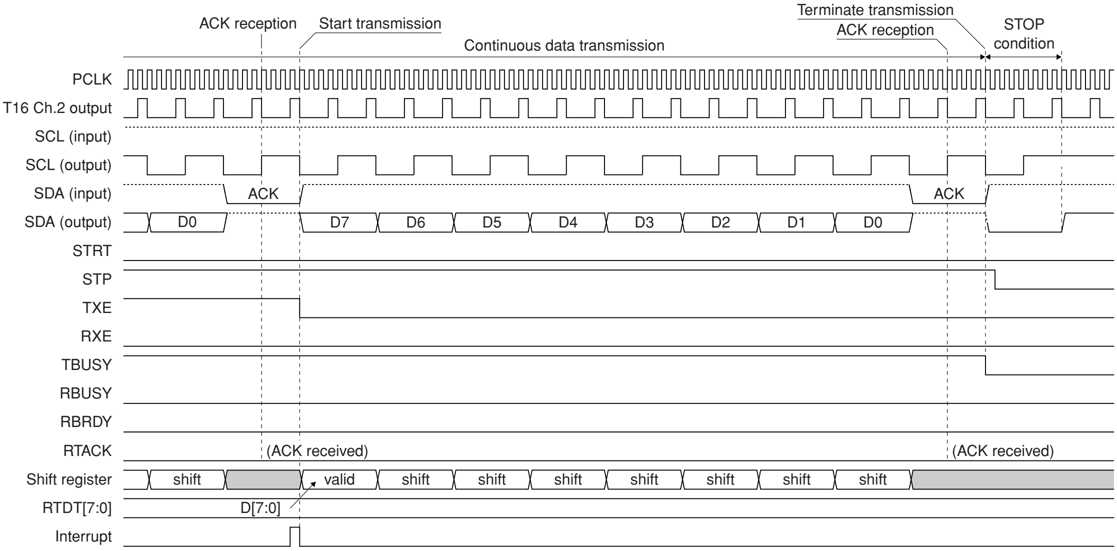


Figure 20.5.6 I^{2}C Timing Chart 2 (data transmission → STOP condition)

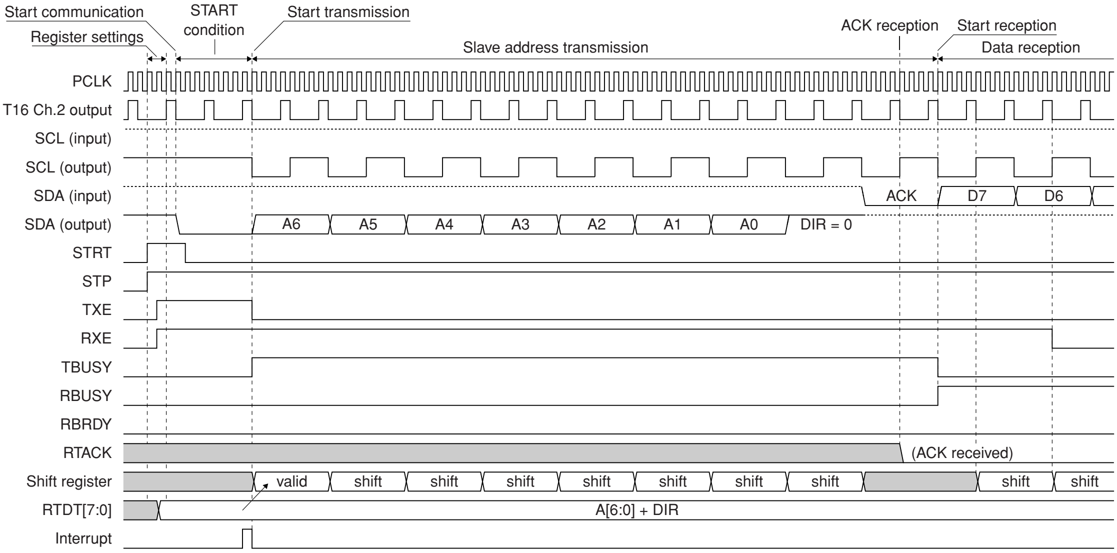


Figure 20.5.7 I^{2}C Timing Chart 3 (START condition → data reception)

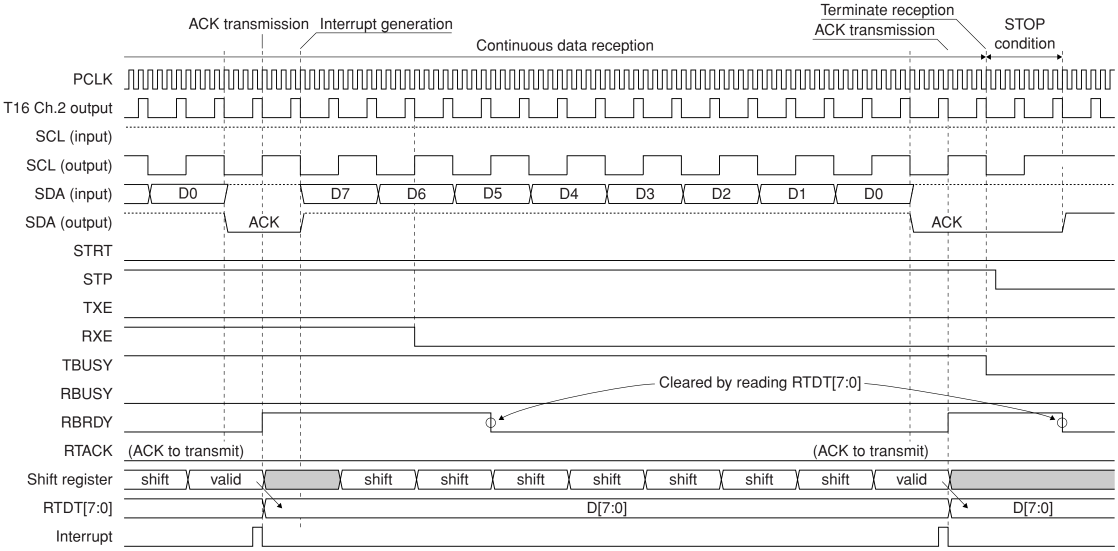


Figure 20.5.8 I²C Timing Chart 4 (data reception → STOP condition)

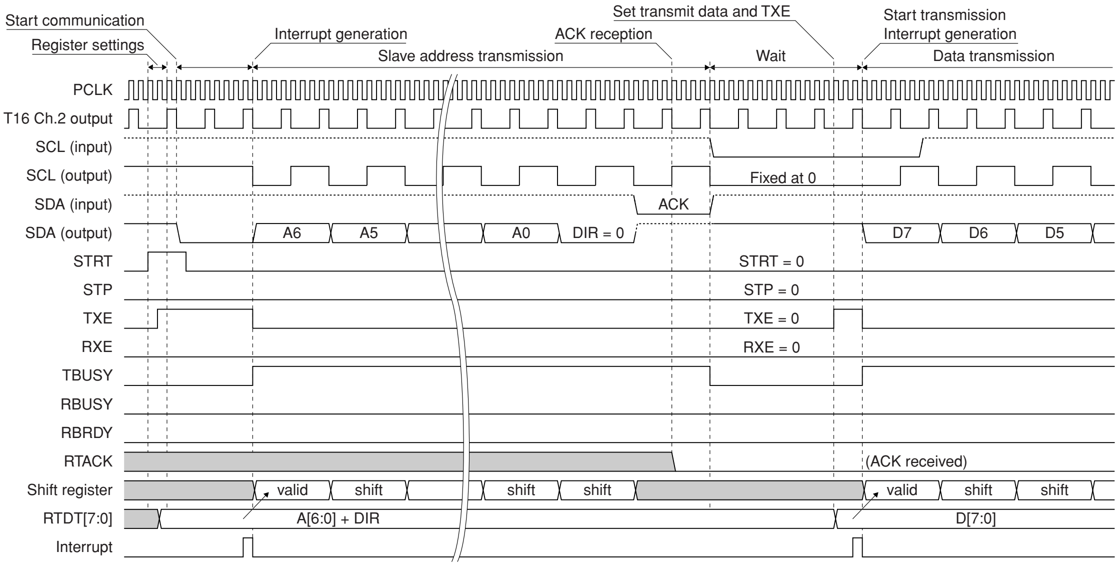


Figure 20.5.9 I²C Timing Chart 5 (wait state)

20.6 I^{2}C Interrupt

The I^{2}C module can generate the following two types of interrupts:

- Transmit buffer empty interrupt
- Receive buffer full interrupt

The I^{2}C module has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with the two causes of interrupt.

Transmit buffer empty interrupt

Set the TINTE bit (D0/I2C_ICTL register) to 1 when using this interrupt. If TINTE is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **TINTE**: Transmit Interrupt Enable Bit in the I^{2}C Interrupt Control (I2C_ICTL) Register (D0/0x4346)

When the transmit data set in RTDT[7:0] (D[7:0]/I2C_DAT register) is transferred to the shift register, the I^{2}C module outputs an interrupt request pulse to the ITC if the transmit buffer empty interrupt has been enabled (TINTE = 1).

∗ **RTDT[7:0]**: Receive/Transmit Data Bits in the I^{2}C Data (I2C_DAT) Register (D[7:0]/0x4344)

If other interrupt conditions are satisfied, an interrupt is generated.

Receive buffer full interrupt

Set the RINTE bit (D1/I2C_ICTL register) to 1 when using this interrupt. If RINTE is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **RINTE**: Receive Interrupt Enable Bit in the I^{2}C Interrupt Control (I2C_ICTL) Register (D1/0x4346)

When data received in the shift register is loaded to RTDT[7:0], the I^{2}C module outputs an interrupt request pulse to the ITC if the receive buffer full interrupt has been enabled (RINTE = 1).

If other interrupt conditions are satisfied, an interrupt is generated.

ITC registers for I^{2}C interrupts

The following shows the control bits of the ITC provided for the I^{2}C module:

Interrupt flag

∗ **IIFT7**: I^{2}C Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D15/0x4300)

Interrupt enable bit

∗ **IIEN7**: I^{2}C Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D15/0x4302)

Interrupt level setup bits

∗ **IILV7[2:0]**: I^{2}C Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV3) Register 3 (D[10:8]/0x4314)

When the I^{2}C outputs an interrupt request pulse, the interrupt flag IIFT7 is set to 1.
If the interrupt enable bit IIEN7 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the I^{2}C interrupt, set the IIEN7 bit to 0.
The IIFT7 flag is always set to 1 by the I^{2}C interrupt request pulse, regardless of how the IIEN7 bit is set (even when set to 0).
The interrupt level setup bits IILV7[2:0] set the interrupt level (0 to 7) of the I^{2}C interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register PSR) in the S1C17 Core is set to 1.
- The I^{2}C interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Chapter 6, "Interrupt Controller (ITC)."

Interrupt vector

The following shows the vector number and vector address for the I^{2}C interrupt:

Vector number: 19 (0x13)
Vector address: 0x804c

20.7 Details of Control Registers

Table 20.7.1 List of I²C Registers

Address	Register name		Function
0x4340	I2C_EN	I²C Enable Register	Enables the I²C module.
0x4342	I2C_CTL	I²C Control Register	Controls the I²C operation and indicates transfer status.
0x4344	I2C_DAT	I²C Data Register	Transmit/receive data
0x4346	I2C_ICTL	I²C Interrupt Control Register	Controls the I²C interrupt.

The following describes each I²C register. These are all 16-bit registers.

Notes:
- When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”
- Be sure to use 16-bit access instructions for reading/writing from/to the I²C registers. The I²C registers do not allow reading/writing using 32-bit and 8-bit access instructions.

0x4340: I^{2}C Enable Register (I2C_EN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^{2}C Enable Register (I2C_EN)	**0x4340** (16 bits)	D15–1	–	reserved	–				–	–	0 when being read.
		D0	**I2CEN**	I^{2}C enable	1	Enable	0	Disable	0	R/W	

D[15:1] Reserved

D0 I2CEN: I^{2}C Enable Bit

Enables/disables operation of the I^{2}C module.

1 (R/W): Enable

0 (R/W): Disable (default)

When I2CEN is set to 1, the I^{2}C module is activated and data transfer is enabled.
When I2CEN is set to 0, the I^{2}C module goes off.

0x4342: I2C Control Register (I2C_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I2C Control Register (I2C_CTL)	0x4342 (16 bits)	D15–10	–	reserved	–				–	–	0 when being read.
		D9	**RBUSY**	Receive busy flag	1	Busy	0	Idle	0	R	
		D8	**TBUSY**	Transmit busy flag	1	Busy	0	Idle	0	R	
		D7–5	–	reserved	–				–	–	0 when being read.
		D4	**NSERM**	Noise remove on/off	1	On	0	Off	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	**STP**	Stop control	1	Stop	0	Ignored	0	R/W	
		D0	**STRT**	Start control	1	Start	0	Ignored	0	R/W	

D[15:10] Reserved

D9 RBUSY: Receive Busy Flag

Indicates the I2C receive operation status.

1 (R): Busy
0 (R): Idle (default)

RBUSY is set to 1 when the I2C starts data reception and stays 1 while data reception is in progress. RBUSY is reset to 0 upon completion of receive operation. Also RBUSY returns to 0 in wait state.

D8 TBUSY: Transmit Busy Flag

Indicates the I2C transmit operation status.

1 (R): Busy
0 (R): Idle (default)

TBUSY is set to 1 when the I2C starts data transmission and stays 1 while data transmission is in progress. TBUSY is reset to 0 upon completion of transmit operation. Also TBUSY returns to 0 in wait state.

D[7:5] Reserved

D4 NSERM: Noise Remove On/Off Bit

Turns the noise remove function on and off.

1 (R/W): On
0 (R/W): Off (default)

The I2C module contains a function to remove noise from the SDA and SCL input signals. This function is enabled by setting NSERM to 1.
Note, however, that the I2C clock (16-bit timer Ch.2 output clock) frequency must be 1/6 of PCLK or lower to use the noise remove function.

D[3:2] Reserved

D1 STP: Stop Control Bit

Generates a STOP condition.

1 (R/W): Generate STOP condition
0 (R/W): Ignore (default)

When STP is set to 1, the I2C module pulls up the SDA line from low to high with the SCL line held at high to generate a STOP condition on the I2C bus. This makes the I2C bus in free status.
Furthermore, the I2C module allows presetting for generating a STOP condition in advance. To do this, set STP to 1 after checking if the I2C is operating (TBUSY = 1 or RBUSY = 1). A STOP condition will be generated upon completion of data transmission/reception (including an ACK transfer).
STP is automatically reset to 0 after a STOP condition has been generated.

D0 STRT: Start Control Bit

Generates a START condition.
1 (R/W): Generate START condition
0 (R/W): Ignore (default)

When STRT is set to 1, the I^{2}C module pulls down the SDA line to low with the SCL line held at high to generate a START condition on the I^{2}C bus. This makes the I^{2}C bus in busy status.
To generate a START condition, set the following registers in the order shown below:

1. Set the slave address to RTDT[7:0] (D[7:0]/I2C_DAT register). (First transmit data when 10-bit address is used; see Figure 20.5.2.)
2. Set TXE (D9/I2C_DAT register) to 1.
3. Set STRT (D0/I2C_CTL register) to 1.

STRT is automatically reset to 0, after a START condition has been generated.

0x4344: I^{2}C Data Register (I2C_DAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^{2}C Data Register (I2C_DAT)	0x4344 (16 bits)	D15–12	–	reserved	–				–	–	0 when being read.
		D11	RBRDY	Receive buffer ready	1	Ready	0	Empty	0	R	
		D10	RXE	Receive execution	1	Receive	0	Ignored	0	R/W	
		D9	TXE	Transmit execution	1	Transmit	0	Ignored	0	R/W	
		D8	RTACK	Receive/transmit ACK	1	Error	0	ACK	0	R/W	
		D7–0	RTDT[7:0]	Receive/transmit data RTDT7 = MSB RTDT0 = LSB	0x0 to 0xff				0x0	R/W	

D[15:12] Reserved

D11 RBRDY: Receive Buffer Ready Flag

Indicates the receive buffer status.

1 (R): Received data is present
0 (R): No received data is present (default)

The RBRDY flag goes 1 when the received data in the shift register is loaded into RTDT[7:0] (D[7:0]) and returns to 0 when the received data is read from RTDT[7:0]. An interrupt can be generated when this flag goes 1. Use this interrupt or read the RBRDY flag to check that RTDT[7:0] contains valid data when reading received data.

D10 RXE: Receive Execution Bit

Execute a data reception for one byte.

1 (R/W): Start data reception
0 (R/W): Ignore (default)

The I^{2}C module starts data reception for one byte by setting RXE to 1 and TXE (D9) to 0. RXE can be set to 1 for the next data reception even if a slave address is being transmitted or data is being received. RXE is reset to 0 when D6 is input to the shift register.

D9 TXE: Transmit Execution Bit

Execute a data transmission for one byte.

1 (R/W): Start data transmission
0 (R/W): Ignore (default)

Set the transmit data to RTDT[7:0] (D[7:0]) and write 1 to TXE to start data transmission. TXE can be set to 1 for the next data transmission even if a slave address or data is being transmitted. TXE is reset to 0 when the data set in RTDT[7:0] is transferred to the shift register.

D8 RTACK: Receive/Transmit ACK Bit

In data transmission

Indicates the acknowledge bit status.

1 (R/W): Error (NACK)
0 (R/W): ACK (default)

This bit is set to 0 when the slave returned ACK after one-byte data has been transmitted. This indicates that the slave could receive the data normally. If this bit is set to 1, the slave may be inactive or it could not receive the data normally.

In data reception

Set the acknowledge bit to be sent to the slave.

1 (R/W): Error (NACK)
0 (R/W): ACK (default)

To return ACK to the slave after data is received, set RTACK to 0 before the I^{2}C module sends the acknowledge bit.
To return NACK, set RTACK to 1.

D[7:0] RTDT[7:0]: Receive/Transmit Data Bits

In data transmission

Set a transmit data in this register. (Default: 0x0)

Data transmission begins when TXE (D9) is set to 1. If a slave address or data is being transmitted, a new transmission starts after the current transmission has completed. The serial-converted data is output from the SDA pin beginning with the MSB, in which the bits set to 0 are output as low-level signals. When the data set in this register is transferred to the shift register, a cause of transmit buffer empty interrupt occurs. The next transmit data can be written to the register after that.

In data reception

The received data can be read from this register. (Default: 0x0)

Data reception begins when RXE (D10) is set to 1. If a slave address is being transmitted or data is being received, a new reception starts after the current reception has completed. When a receive operation is completed and the data received in the shift register is loaded to this register, the RBRDY flag (D11) is set and a cause of receive buffer full interrupt occurs. Thereafter, the data can be read out at any time before a receive operation for the next data is completed.

If the next data receive operation is completed before this register is read out, the data in it is overwritten with the newly received data.

The serial data input from the SDA pin is converted into parallel data beginning with the MSB, with the high-level signals changed to 1s and the low-level signals changed to 0s. The resulting data is stored in this register.

0x4346: I²C Interrupt Control Register (I2C_ICTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Interrupt Control Register (I2C_ICTL)	**0x4346** (16 bits)	D15–2	**–**	reserved	–				–	–	0 when being read.
		D1	**RINTE**	Receive interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**TINTE**	Transmit interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:2] Reserved

D1 RINTE: Receive Interrupt Enable Bit

Enables/disables the I²C receive buffer full interrupt.

1 (R/W): Enable

0 (R/W): Disable (default)

When RINTE is set to 1, I²C receive buffer full interrupt requests to the ITC are enabled. A receive buffer full interrupt request occurs when the data received in the shift register is loaded to RTDT[7:0] (D[7:0]/I2C_DAT register) (receive operation completed).

When RINTE is set to 0, an I²C receive buffer full interrupt is not generated.

D0 TINTE:Transmit Interrupt Enable Bit

Enables/disables the I²C transmit buffer empty interrupt.

1 (R/W): Enable

0 (R/W): Disable (default)

When TINTE is set to 1, I²C transmit buffer empty interrupt requests to the ITC are enabled. A transmit buffer empty interrupt request occurs when the data written to RTDT[7:0] (D[7:0]/I2C_DAT register) is transferred to the shift register.

When TINTE is set to 0, an I²C transmit buffer empty interrupt is not generated.

THIS PAGE IS BLANK.

21 Remote Controller (REMC)

21.1 Outline of the REMC

The S1C17701 is equipped with a remote controller (REMC) module for generating/receiving infrared remote control signals. The REMC module consists of a carrier generator for generating a carrier signal using a prescaler output clock, an 8-bit down counter for counting the transmit/receive data length, a modulator for generating transmit data with a designated carrier length, and an edge detector for detecting rising and falling edges from the input signal.

Also the REMC module can generate three types of interrupts: counter underflow interrupt to notify that a transmission for designated data length has finished, and input rising edge detection and falling edge detection interrupts for data receive processing.

Figure 21.1.1 shows the structure of the REMC module.

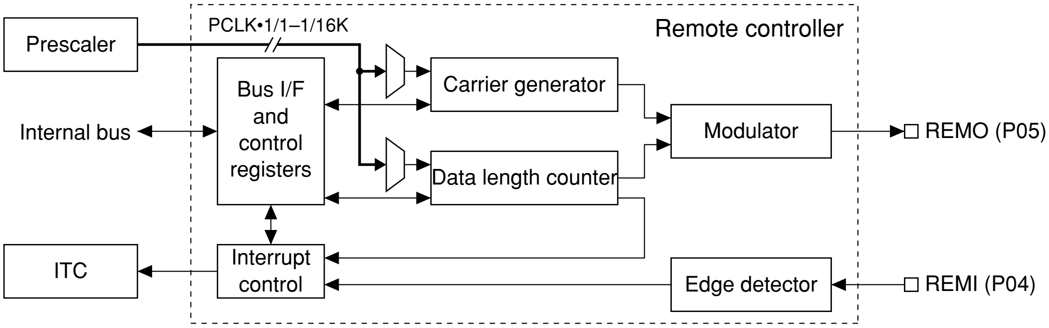


Figure 21.1.1 Structure of REMC Module

21.2 REMC I/O Pins

Table 21.2.1 lists the REMC input/output pins.

Table 21.2.1 List of REMC Pins

Pin name	I/O	Size	Function
REMI (P04)	I	1	Remote controller receive data input pin This pin inputs receive data.
REMO (P05)	O	1	Remote controller transmit data output pin This pin outputs the modulated remote control transmit data.

The REMC input/output pins (REMI, REMO) are shared with the I/O ports (P04, P05) and they are initialized as general-purpose I/O port pins by default. Before using these pins for the REMC, the pin functions must be switched using the P0_PMUX register. Set the control bits shown below to 1 to configure the pins for the REMC.

P04 → REMI

∗ **P04MUX**: P04 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D4/0x52a0)

P05 → REMO

∗ **P05MUX**: P05 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D5/0x52a0)

For details on switching pin function, see Section 10.2, "Selecting I/O Pin Functions (Port MUX)."

21.3 Carrier Generator

The REMC module contains a carrier generator that generates a transmit carrier signal according to the clock, H carrier length, and L carrier length set with software.

A prescaler output clock is used to generate the carrier signal. Use the CGCLK[3:0] bits (D[7:4]/REMC_PSC register) to select one from the 15 clocks, PCLK divided by 1 to PCLK divided by 16K, generated by the prescaler.

* **CGCLK[3:0]**: Carrier Generator Clock Select Bits in the REMC Prescaler Clock Select (REMC_PSC) Register (D[7:4]/0x5341)

Table 21.3.1 Selecting a Clock for Carrier Generator

CGCLK[3:0]	Prescaler output clock	CGCLK[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK•1/128
0xe	PCLK•1/16384	0x6	PCLK•1/64
0xd	PCLK•1/8192	0x5	PCLK•1/32
0xc	PCLK•1/4096	0x4	PCLK•1/16
0xb	PCLK•1/2048	0x3	PCLK•1/8
0xa	PCLK•1/1024	0x2	PCLK•1/4
0x9	PCLK•1/512	0x1	PCLK•1/2
0x8	PCLK•1/256	0x0	PCLK•1/1

(Default: 0x0)

For controlling the prescaler, see Chapter 9, "Prescaler (PSC)."

Note: Before the REMC module can be used, the prescaler must be run.

The lengths of H period and L period of the carrier signal can be set using REMCH[5:0] (D[5:0]/REMC_CARH register) and REMCL[5:0] (D[5:0]/REMC_CARL register). The H/L period lengths should be set as the number of clock (selected as above) cycles +1.

* **REMCH[5:0]**: H Carrier Length Setup Bits in the REMC H Carrier Length Setup (REMC_CARH) Register (D[5:0]/0x5342)
* **REMCL[5:0]**: L Carrier Length Setup Bits in the REMC L Carrier Length Setup (REMC_CARL) Register (D[5:0]/0x5343)

The H and L carrier lengths are calculated by the expressions below.

$$\text{H carrier length} = \frac{\text{REMCH} + 1}{\text{clk\_in}} \text{ [s]}$$

$$\text{L carrier length} = \frac{\text{REMCL} + 1}{\text{clk\_in}} \text{ [s]}$$

REMCH: H carrier length register data
REMCL: L carrier length register data
clk_in: Prescaler output clock frequency

The carrier signal is generated according to these settings as shown in Figure 21.3.1.

Example: CGCLK[3:0] = 0x2 (PCLK•1/4), REMCH[5:0] = 2, REMCL[5:0] = 1

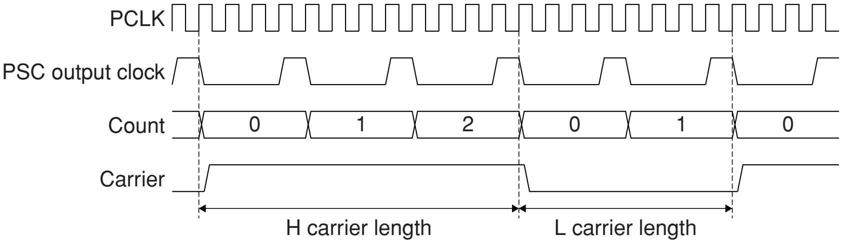


Figure 21.3.1 Carrier Signal Generation

21.4 Setting Clock for Data Length Counter

The data length counter is an 8-bit down counter for setting the data length for data transmission.
During data transmission, write a value equivalent to the data pulse width in this counter. The counter starts counting down from the set value and stops after generating a cause of underflow interrupt when the counter reaches 0. The next transmit data can be set using this interrupt.

This counter is also used to measure the receive data length during data reception. When data is received, an interrupt can be generated at the rising edge and falling edge of the input signal. Use an input transition interrupt to set the data length counter to 0xff and read the counter value when the next interrupt caused by an input transition occurs. The input data pulse width can be obtained from the difference between 0xff and the read value.

As in the case of the carrier generator, the data length counter uses a prescaler output clock as the count clock. Use the LCCLK[3:0] bits (D[3:0]/REMC_PSC register) provided separately with the carrier generator to select one of the 15 prescaler output clocks.

∗ **LCCLK[3:0]**: Length Counter Clock Select Bits in the REMC Prescaler Clock Select (REMC_PSC) Register (D[3:0]/0x5341)

Table 21.4.1 Selecting a Data Length Counter Clock

LCCLK[3:0]	Prescaler output clock	LCCLK[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK•1/128
0xe	PCLK•1/16384	0x6	PCLK•1/64
0xd	PCLK•1/8192	0x5	PCLK•1/32
0xc	PCLK•1/4096	0x4	PCLK•1/16
0xb	PCLK•1/2048	0x3	PCLK•1/8
0xa	PCLK•1/1024	0x2	PCLK•1/4
0x9	PCLK•1/512	0x1	PCLK•1/2
0x8	PCLK•1/256	0x0	PCLK•1/1

(Default: 0x0)

The data length counter can count up to 256. Select a count clock so that the data length can be counted within this range.

21.5 Controlling Data Transmission/Reception

Before starting data transfer, set up the conditions by the procedure below.

(1) Configure the carrier signal. See Section 21.3.
(2) Select a clock for the data length counter. See Section 21.4.
(3) Set up the interrupt conditions. See Section 21.6.

Note: Make sure that the REMC module is disabled (REMEN/REMC_CFG register = 0) before setting the conditions above.

∗ **REMEN**: REMC Enable Bit in the REMC Configuration (REMC_CFG) Register (D0/0x5340)

Data transmit control

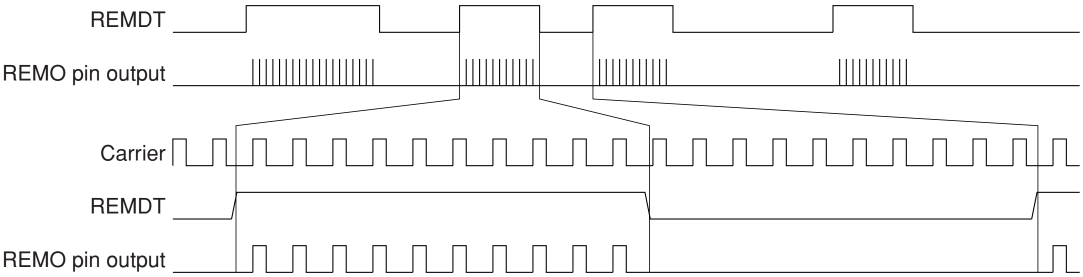


Figure 21.5.1 Data Transmission

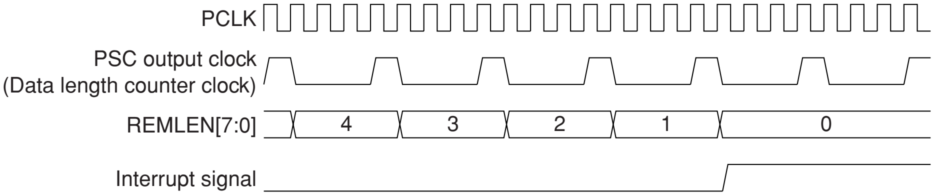


Figure 21.5.2 Underflow Interrupt Generation Timing

(1) Setting data transmit mode

Write 0 to REMMD (D1/REMC_CFG register) to set the REMC in data transmit mode.

∗ **REMMD**: REMC Mode Select Bit in the REMC Configuration (REMC_CFG) Register (D1/0x5340)

(2) Enabling data transmission

Set REMEN (D0/REMC_CFG register) to 1 to enable the REMC operation. This makes the REMC start the data transmit operation.

To prevent unnecessary data from being transmitted, set REMDT (D0/REMC_ST register) to 0 and REMLEN[7:0] (D[7:0]/REMC_LCNT register) to 0x0 before writing 1 to REMEN.

(3) Setting transmit data

Set data to be transmitted (high or low) to REMDT (D0/REMC_ST register).

∗ **REMDT**: Transmit/Receive Data Bit in the REMC Status (REMC_ST) Register (D0/0x5344)

Setting REMDT to 1 specifies high output; setting it to 0 specifies low output. The specified data is modulated with the carrier signal and output from the REMO pin.

(4) Setting the data pulse width

Write the value equivalent to the pulse width (high period or low period) of the transmit data to REMLEN[7:0] (D[7:0]/REMC_LCNT register) to set it to the data length counter.

∗ **REMLEN[7:0]**: Transmit/Receive Data Length Count Bits in the REMC Length Counter (REMC_LCNT) Register (D[7:0]/0x5345)

Use the following equation to determine the value to be set to the data length counter.

Set value = Data pulse width (second) × Prescaler output clock frequency (Hz)

The data length counter starts counting down from the written value using the selected prescaler output clock. When the data counter value reaches 0, a cause of underflow interrupt occurs. The REMC module sends an interrupt request to the interrupt controller (ITC) if the interrupt is enabled. The data length counter stops counting at this point.

(5) Interrupt handling

To continue data transmission, set the next transmit data (3) and data pulse width (4) in the interrupt handler routine executed because of a data length counter underflow.

(6) Terminating data transmission

After the last data transfer has finished (after an underflow interrupt occurs), write 0 to the REMEN bit to terminate data transmission.

Data receive control

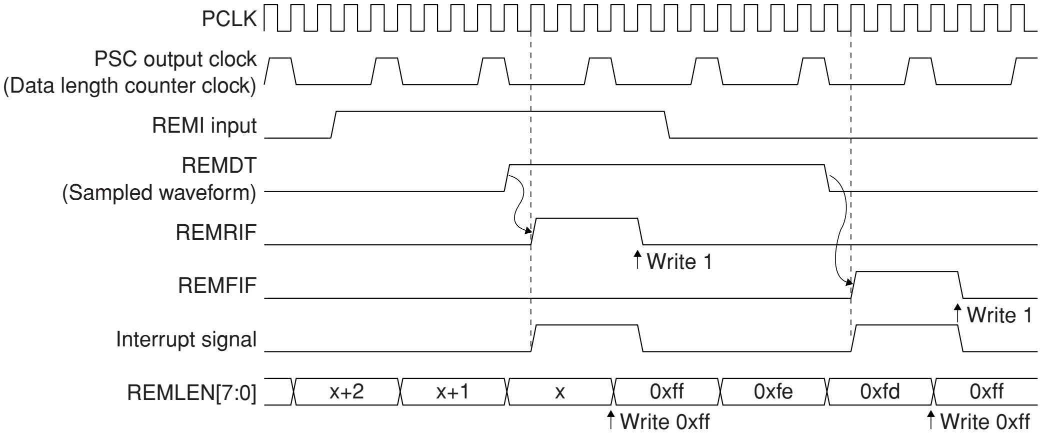


Figure 21.5.3 Data Reception

(1) Setting data receive mode

Write 1 to REMMD (D1/REMC_CFG register) to set the REMC in data receive mode.

(2) Enabling data reception

Set REMEN (D0/REMC_CFG register) to 1 to enable the REMC operation. This makes the REMC start the data receive operation (input signal edge detection).

The REMC module samples the signal input to the REMI pin with the prescaler output clock selected for the carrier generator to detect input transition (rising edge or falling edge of the signal). When a signal edge is detected, a cause of rising edge or falling edge interrupt occurs and the REMC module sends an interrupt request to the ITC if the interrupt is enabled. The rising edge and falling edge interrupts can be enabled individually.

Note that a signal transition is regarded as noise if the signal level after the input changes is not sampled for two or more sampling clock cycles. In this case no rising edge or falling edge interrupt occurs.

(3) Interrupt handling

When a rising edge or falling edge interrupt occurs, write 0xff to REMLEN[7:0] (D[7:0]/REMC_LCNT register) to set it to the data length counter in the interrupt handler routine.

The data length counter starts counting with the selected prescaler output clock from the value written to it.

The received data can be read from REMDT (D0/REMC_ST register).

At the trailing edge of the data pulse, the next falling edge or rising edge interrupt occurs. Read the data length counter at that point. The data length can be calculated from the difference between 0xff and the read value.
To continue data reception, set the data length counter to 0xff again and wait until the next interrupt.

If the data length counter reaches 0 without an interrupt generated after the counter is set to 0xff, either no receive data remains or a receive error has occurred. A data length counter underflow interrupt occurs even in data reception, use it for a terminate/error processing.

(4) Terminating data reception

After the last data transfer has finished, write 0 to the REMEN bit to terminate data reception.

21.6 REMC Interrupt

The REMC module can generate the following three types of interrupts:

- Underflow interrupt
- Rising edge interrupt
- Falling edge interrupt

The REMC module has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with the three causes of interrupt. To determine the cause of interrupt that has occurred, read the interrupt flags in the REMC module.

Underflow interrupt

This interrupt request occurs when the data length counter reaches 0 during count-down, and it sets the interrupt flag REMUIF (D0/REMC_IFLG register) in the REMC module to 1.
During data transmission, this interrupt notifies the application program that a data transmission with the data length specified has completed. During data reception, this interrupt notifies the application program that a data reception has completed or a receive error has occurred.

∗ **REMUIF**: Underflow Interrupt Flag in the REMC Interrupt Flag (REMC_IFLG) Register (D0/0x5347)

Set the REMUIE bit (D0/REMC_IMSK register) to 1 when using this interrupt. If REMUIE is set to 0 (default), REMUIF will not be set to 1 and an interrupt request by this cause will not be sent to the ITC.

∗ **REMUIE**: Underflow Interrupt Enable Bit in the REMC Interrupt Mask (REMC_IMSK) Register (D0/0x5346)

If REMUIF is set to 1, the REMC module outputs the interrupt request signal to the ITC. The interrupt request signal sets the REMC interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The REMC interrupt handler routine should read the REMUIF flag to check if the interrupt has occurred due to a data length counter underflow or another cause.
Furthermore, the interrupt handler routine must reset (write 1 to) REMUIF in the REMC module as well as the REMC interrupt flag in the ITC, to clear the cause of interrupt.

Rising edge interrupt

This interrupt request occurs when the signal input to the REMI pin goes high from low status, and it sets the interrupt flag REMRIF (D1/REMC_IFLG register) in the REMC module to 1.
During data reception, run the data length counter between this interrupt and the corresponding falling edge interrupt. The receive data pulse width can be calculated from the count value.

∗ **REMRIF**: Rising Edge Interrupt Flag in the REMC Interrupt Flag (REMC_IFLG) Register (D1/0x5347)

Set the REMRIE bit (D1/REMC_IMSK register) to 1 when using this interrupt. If REMRIE is set to 0 (default), REMRIF will not be set to 1 and an interrupt request by this cause will not be sent to the ITC.

∗ **REMRIE**: Rising Edge Interrupt Enable Bit in the REMC Interrupt Mask (REMC_IMSK) Register (D1/0x5346)

If REMRIF is set to 1, the REMC module outputs the interrupt request signal to the ITC. The interrupt request signal sets the REMC interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The REMC interrupt handler routine should read the REMRIF flag to check if the interrupt has occurred due to detection of a rising edge of the input signal or another cause.
Furthermore, the interrupt handler routine must reset (write 1 to) REMRIF in the REMC module as well as the REMC interrupt flag in the ITC, to clear the cause of interrupt.

Falling edge interrupt

This interrupt request occurs when the signal input to the REMI pin goes low from high status, and it sets the interrupt flag REMFIF (D2/REMC_IFLG register) in the REMC module to 1.
During data reception, run the data length counter between this interrupt and the corresponding rising edge interrupt. The receive data pulse width can be calculated from the count value.

∗ **REMFIF**: Falling Edge Interrupt Flag in the REMC Interrupt Flag (REMC_IFLG) Register (D2/0x5347)

Set the REMFIE bit (D2/REMC_IMSK register) to 1 when using this interrupt. If REMFIE is set to 0 (default), REMFIF will not be set to 1 and an interrupt request by this cause will not be sent to the ITC.

∗ **REMFIE**: Falling Edge Interrupt Enable Bit in the REMC Interrupt Mask (REMC_IMSK) Register (D2/0x5346)

If REMFIF is set to 1, the REMC module outputs the interrupt request signal to the ITC. The interrupt request signal sets the REMC interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The REMC interrupt handler routine should read the REMFIF flag to check if the interrupt has occurred due to detection of a falling edge of the input signal or another cause.
Furthermore, the interrupt handler routine must reset (write 1 to) REMFIF in the REMC module as well as the REMC interrupt flag in the ITC, to clear the cause of interrupt.

ITC registers for REMC interrupt

When a cause of interrupt that has been enabled occurs according to the interrupt condition settings shown above, the REMC module asserts the interrupt signal sent to the ITC. To generate a REMC interrupt, set the interrupt level and enable the interrupt using the ITC registers.
The following shows the control bits for the REMC interrupt in the ITC.

Interrupt flag in the ITC

∗ **IIFT5**: Remote Controller Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D13/0x4300)

Interrupt enable bit in the ITC

∗ **IIEN5**: Remote Controller Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D13/0x4302)

Interrupt level setup bits in the ITC

∗ **IILV5[2:0]**: REMC Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV2) Register 2 (D[10:8]/0x4312)

The interrupt signal sent from the REMC module sets IIFT5 to 1. If IIEN5 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the REMC interrupt, set IIEN5 to 0. IIFT5 is always set to 1 by the interrupt signal sent from the REMC module, regardless of how IIEN5 is set (even when set to 0).
IILV5[2:0] sets the interrupt level (0 to 7) of the REMC interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The REMC interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Chapter 6, “Interrupt Controller (ITC).”

Interrupt vector

The following shows the vector number and vector address for the REMC interrupt:

Vector number: 17 (0x11)
Vector address: 0x8044

21.7 Details of Control Registers

Table 21.7.1 List of REMC Registers

Address	Register name		Function
0x5340	REMC_CFG	REMC Configuration Register	Selects/enables transmission/reception.
0x5341	REMC_PSC	REMC Prescaler Clock Select Register	Selects a prescaler output clock.
0x5342	REMC_CARH	REMC H Carrier Length Setup Register	Sets up the H period of the carrier.
0x5343	REMC_CARL	REMC L Carrier Length Setup Register	Sets up the L period of the carrier.
0x5344	REMC_ST	REMC Status Register	Transmit/receive bit
0x5345	REMC_LCNT	REMC Length Counter Register	Sets the transmit/receive data length.
0x5346	REMC_IMSK	REMC Interrupt Mask Register	Enables/disables interrupt.
0x5347	REMC_IFLG	REMC Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The following describes each REMC register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5340: REMC Configuration Register (REMC_CFG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
REMC Configuration Register (REMC_CFG)	**0x5340** (8 bits)	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**REMMD**	REMC mode select	1	Receive	0	Transmit	0	R/W	
		D0	**REMEN**	REMC enable	1	Enable	0	Disable	0	R/W	

D[7:2] Reserved

D1 REMMD: REMC Mode Select Bit

Selects the data transmit/receive direction.

1 (R/W): Reception

0 (R/W): Transmission (default)

D0 REMEN: REMC Enable Bit

Enables or disables the REMC module to transmit/receive data.

1(R/W): Enable

0(R/W): Disable (default)

When REMEN is set to 1, the REMC module starts data transmission or data reception according to the REMMD (D1) setting. When REMEN is set to 0, the REMC module stops operating.

0x5341: REMC Prescaler Clock Select Register (REMC_PSC)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
REMC Prescaler Clock Select Register (REMC_PSC)	**0x5341** (8 bits)	D7–4	**CGCLK[3:0]**	Carrier generator clock select (Prescaler output clock)	CGCLK[3:0] LCCLK[3:0] 0xf 0xe 0xd 0xc 0xb 0xa 0x9 0x8 0x7 0x6 0x5 0x4 0x3 0x2 0x1 0x0	Clock reserved PCLK•1/16384 PCLK•1/8192 PCLK•1/4096 PCLK•1/2048 PCLK•1/1024 PCLK•1/512 PCLK•1/256 PCLK•1/128 PCLK•1/64 PCLK•1/32 PCLK•1/16 PCLK•1/8 PCLK•1/4 PCLK•1/2 PCLK•1/1	0x0	R/W	
		D3–0	**LCCLK[3:0]**	Length counter clock select (Prescaler output clock)			0x0	R/W	

D[7:4] CGCLK[3:0]: Carrier Generator Clock Select Bits

These bits select the carrier generator clock from 15 prescaler output clocks.

Table 21.7.2 Selecting the Carrier Generator Clock

CGCLK[3:0]	Prescaler output clock	CGCLK[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK•1/128
0xe	PCLK•1/16384	0x6	PCLK•1/64
0xd	PCLK•1/8192	0x5	PCLK•1/32
0xc	PCLK•1/4096	0x4	PCLK•1/16
0xb	PCLK•1/2048	0x3	PCLK•1/8
0xa	PCLK•1/1024	0x2	PCLK•1/4
0x9	PCLK•1/512	0x1	PCLK•1/2
0x8	PCLK•1/256	0x0	PCLK•1/1

(Default: 0x0)

D[3:0] LCCLK[3:0]: Length Counter Clock Select Bits

These bits select the data length counter clock from 15 prescaler output clocks.

Table 21.7.3 Selecting the Data Length Counter Clock

LCCLK[3:0]	Prescaler output clock	LCCLK[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK•1/128
0xe	PCLK•1/16384	0x6	PCLK•1/64
0xd	PCLK•1/8192	0x5	PCLK•1/32
0xc	PCLK•1/4096	0x4	PCLK•1/16
0xb	PCLK•1/2048	0x3	PCLK•1/8
0xa	PCLK•1/1024	0x2	PCLK•1/4
0x9	PCLK•1/512	0x1	PCLK•1/2
0x8	PCLK•1/256	0x0	PCLK•1/1

(Default: 0x0)

Note: When setting the clocks, make sure the REMC module is idle (REMEN/REMC_CFG register = 0).

0x5342: REMC H Carrier Length Setup Register (REMC_CARH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC H Carrier Length Setup Register (REMC_CARH)	**0x5342** (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	**REMCH[5:0]**	H carrier length setup	0x0 to 0x3f	0x0	R/W	

D[7:6] Reserved

D[5:0] REMCH[5:0]: H Carrier Length Setup Bits

Sets the H period length of the carrier signal. (Default: 0x0)

The H period length should be set as the number of carrier generator clock cycles +1. The carrier generator clock is selected with CGCLK[3:0] (D[7:4]/REMC_PSC register).

The H carrier length is calculated by the expression below.

$$\text{H carrier length} = \frac{\text{REMCH} + 1}{\text{clk\_in}} \text{ [s]}$$

REMCH: H carrier length register data
clk_in: Prescaler output clock frequency

The L period length is specified with REMCL[5:0] (D[5:0]/REMC_CARL register).

The carrier signal is generated according to these settings as shown in Figure 21.7.1.

Example: CGCLK[3:0] = 0x2 (PCLK•1/4), REMCH[5:0] = 2, REMCL[5:0] = 1

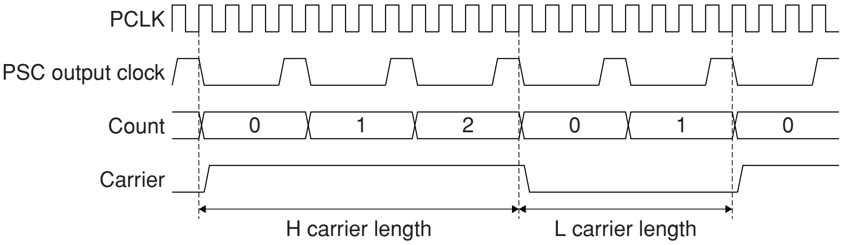


Figure 21.7.1 Carrier Signal Generation

0x5343: REMC L Carrier Length Setup Register (REMC_CARL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC L Carrier Length Setup Register (REMC_CARL)	**0x5343** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**REMCL[5:0]**	L carrier length setup	0x0 to 0x3f	0x0	R/W	

D[7:6] Reserved

D[5:0] REMCL[5:0]: L Carrier Length Setup Bits

Sets the L period length of the carrier signal. (Default: 0x0)

The L period length should be set as the number of carrier generator clock cycles +1. The carrier generator clock is selected with CGCLK[3:0] (D[7:4]/REMC_PSC register).

The L carrier length is calculated by the expression below.

$$\text{L carrier length} = \frac{\text{REMCL} + 1}{\text{clk\_in}} \text{ [s]}$$

REMCL: L carrier length register data
clk_in: Prescaler output clock frequency

The H period length is specified with REMCH[5:0] (D[5:0]/REMC_CARH register).

The carrier signal is generated according to these settings as shown in Figure 21.7.1.

0x5344: REMC Status Register (REMC_ST)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Status Register (REMC_ST)	**0x5344** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**REMDT**	Transmit/receive data	1 1 (H) 0 0 (L)	0	R/W	

D[7:1] Reserved

D0 REMDT: Transmit/Receive Data Bit

Set transmit data during data transmission. Read the received data from this bit during data reception.

1 (R/W): 1 (H)

0 (R/W): 0 (L) (default)

When REMEN (D0/REMC_CFG register) is set to 1, the REMC module modulates the REMDT set value with the carrier signal and outputs the modulated signal from the REMO pin during data transmission. During data reception, the signal level of the input data pulse is set to this bit.

0x5345: REMC Length Counter Register (REMC_LCNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Length Counter Register (REMC_LCNT)	**0x5345** (8 bits)	D7–0	**REMLEN[7:0]**	Transmit/receive data length count (down counter)	0x0 to 0xff	0x0	R/W	

D[7:0] REMLEN[7:0]: Transmit/Receive Data Length Count Bits

Sets the data length counter value to start the counter. (Default: 0x0)

The counter stops when it reaches 0 and generates a cause of underflow interrupt.

During data transmission

During data transmission, set the transmit data.

By writing a value equivalent to the data pulse width to this register, the data length counter starts counting down from the set value and stops after generating a cause of underflow interrupt when the counter reaches 0. The next transmit data can be set using this interrupt.

During data reception

During data reception, an interrupt can be generated at the rising edge and falling edge of the input signal. Use an input transition interrupt to set the data length counter to 0xff and read the counter value when the next interrupt caused by an input transition occurs. The input data pulse width can be obtained from the difference between 0xff and the read value.

0x5346: REMC Interrupt Mask Register (REMC_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
REMC Interrupt Mask Register (REMC_IMSK)	**0x5346** (8 bits)	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**REMFIE**	Falling edge interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**REMRIE**	Rising edge interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**REMUIE**	Underflow interrupt enable	1	Enable	0	Disable	0	R/W	

This register enables or disables the interrupt requests caused by a data length counter underflow, the rising edge of the input signal, and the falling edge of the input signal, individually. Setting an interrupt enable bit to 1 enables the interrupt request by the corresponding cause of interrupt; setting it to 0 disables the interrupt.

In addition, it is necessary to set the REMC interrupt enable bit in the ITC to interrupt enabled to actually generate an interrupt.

D[7:3] Reserved

D2 REMFIE: Falling Edge Interrupt Enable Bit

Enables or disables the interrupt by detecting the falling edge of the input signal.

1 (R/W): Enable interrupt
0 (R/W): Disable interrupt (default)

D1 REMRIE: Rising Edge Interrupt Enable Bit

Enables or disables the interrupt by detecting the rising edge of the input signal.

1 (R/W): Enable interrupt
0 (R/W): Disable interrupt (default)

D0 REMUIE: Underflow Interrupt Enable Bit

Enables or disables the interrupt by a data length counter underflow.

1 (R/W): Enable interrupt
0 (R/W): Disable interrupt (default)

0x5347: REMC Interrupt Flag Register (REMC_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
REMC Interrupt Flag Register (REMC_IFLG)	**0x5347** (8 bits)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	**REMFIF**	Falling edge interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	**REMRIF**	Rising edge interrupt flag					0	R/W	
		D0	**REMUIF**	Underflow interrupt flag					0	R/W	

This register indicates whether causes of data length counter underflow, input signal rising edge, and input signal falling edge interrupts have occurred or not. When a REMC interrupt occurs, read the interrupt flags in this register to determine the cause of interrupt that has occurred.

The interrupt flags are set to 1 when the data length counter underflows, a rising edge of the input signal is detected or a falling edge of the input signal is detected if the corresponding interrupt enable bit has been set to 1. At the same time, the REMC interrupt request signal is output to the ITC. The interrupt request signal sets the REMC interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the interrupt flag before the REMC interrupt is enabled using the interrupt enable bit.

D[7:3] Reserved

D2 REMFIF: Falling Edge Interrupt Flag

This is the interrupt flag to indicate the falling edge interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

REMFIF is set to 1 at the falling edge of the input signal only when REMFIE (D2/REMC_IMSK register) is set to 1.

D1 REMRIF: Rising Edge Interrupt Flag

This is the interrupt flag to indicate the rising edge interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

REMRIF is set to 1 at the rising edge of the input signal only when REMRIE (D1/REMC_IMSK register) is set to 1.

D0 REMUIF: Underflow Interrupt Flag

This is the interrupt flag to indicate the underflow interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

REMUIF is set to 1 when the data length counter underflows only when REMUIE (D0/REMC_IMSK register) is set to 1.

21.8 Precaution

Before the REMC module can start operating, the prescaler must be run.

THIS PAGE IS BLANK.

22 LCD Driver (LCD)

22.1 Configuration of LCD Driver

The S1C17701 is equipped with an LCD driver which can drive an LCD panel with a maximum of 1,792 pixels (56 segments × 32 commons).

Figure 22.1.1 shows the configuration of the LCD driver and the drive power supply.

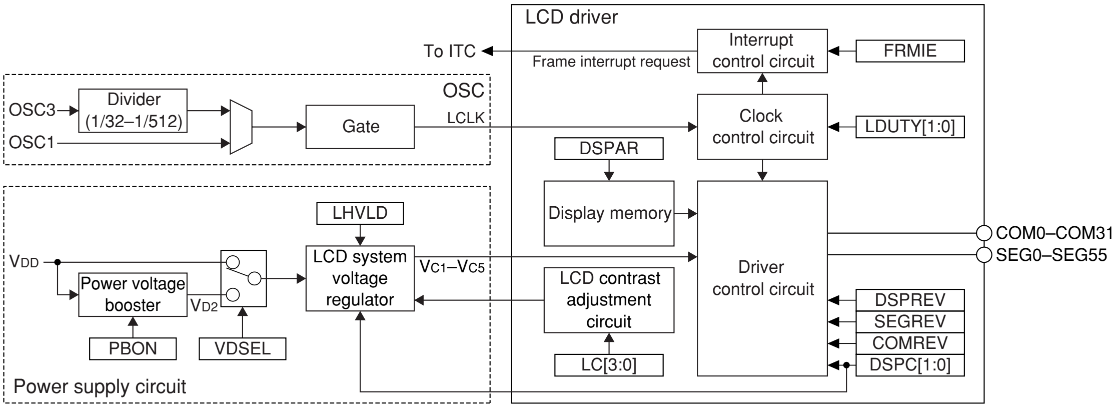


Figure 22.1.1 Configuration of LCD Driver and Drive Power Supply

22.2 LCD Power Supply

The S1C17701 generates the LCD drive voltages V_{C1} to V_{C5} using the on-chip LCD system voltage regulator and power voltage booster. It is not necessary to supply external voltage. For details of the LCD power supply, see Chapter 4, “Power Supply.”

22.3 LCD Clock

Figure 22.3.1 shows the LCD clock supply system.

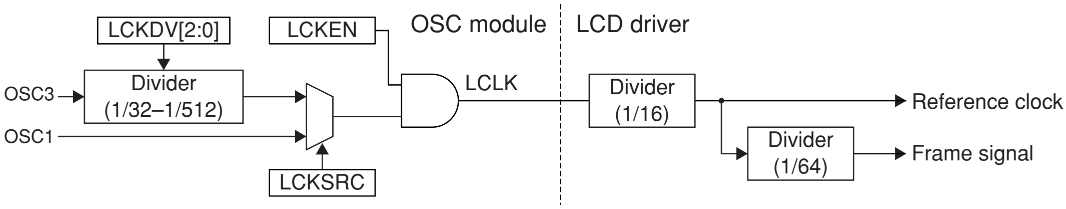


Figure 22.3.1 LCD Clock System

22.3.1 LCD Operating Clock

The LCD operating clock (LCLK) is generated in the LCD clock generator in the OSC module. For details of the OSC module, see Chapter 7, “Oscillator (OSC).”

22.3.2 Frame Signal

The frame signal is generated by dividing LCLK by 1024. The frame frequency is determined as follows. One frame period shown in Figures 22.4.1 and 22.4.2 is assumed as a frame frequency.

When OSC1 (32.768 kHz typ.) is selected as the clock source

Frame frequency = 64 Hz (typ.)

When OSC3 is selected as the clock source

$$\text{Frame frequency} = \frac{f_{OSC3}}{512} \times \text{LCKDV [Hz]}$$

f_{OSC3}: OSC3 clock frequency [Hz]
LCKDV: OSC3 division ratio 1/32 to 1/512

22.4 Switching Drive Duty

The drive duty can be selected from 1/32 and 1/16 using LDUTY[1:0] (D[1:0]/LCD_CCTL register). Table 22.4.1 shows the relationship of the LDUTY[1:0] setting, drive duty and the maximum number of display pixels.

∗ **LDUTY[1:0]**: LCD Duty Select Bits in the LCD Clock Control (LCD_CCTL) Register (D[1:0]/0x50a2)

Table 22.4.1 Setting the Drive Duty

LDUTY[1:0]	Duty	Effective COM pins	Effective SEG pins	Max number of pixels
0x3	Reserved	–	–	–
0x2	1/32	COM0–COM31	SEG0–SEG55	1,792 pixels
0x1	1/16	COM0–COM15	SEG0–SEG71	1,152 pixels
0x0	Reserved	–	–	–

(Default: 0x2)

The COM16–COM31/SEG71–SEG56 pins are configured as common output pins when 1/32 duty is selected or configured as segment output pins when 1/16 duty is selected.

The drive bias is fixed at 1/5 (using five voltages, V_{C1}, V_{C2}, V_{C3}, V_{C4} and V_{C5}) regardless of the drive duty selected. Figures 22.4.1 and 22.4.2 show the drive waveforms for 1/32 duty and 1/16 duty, respectively.

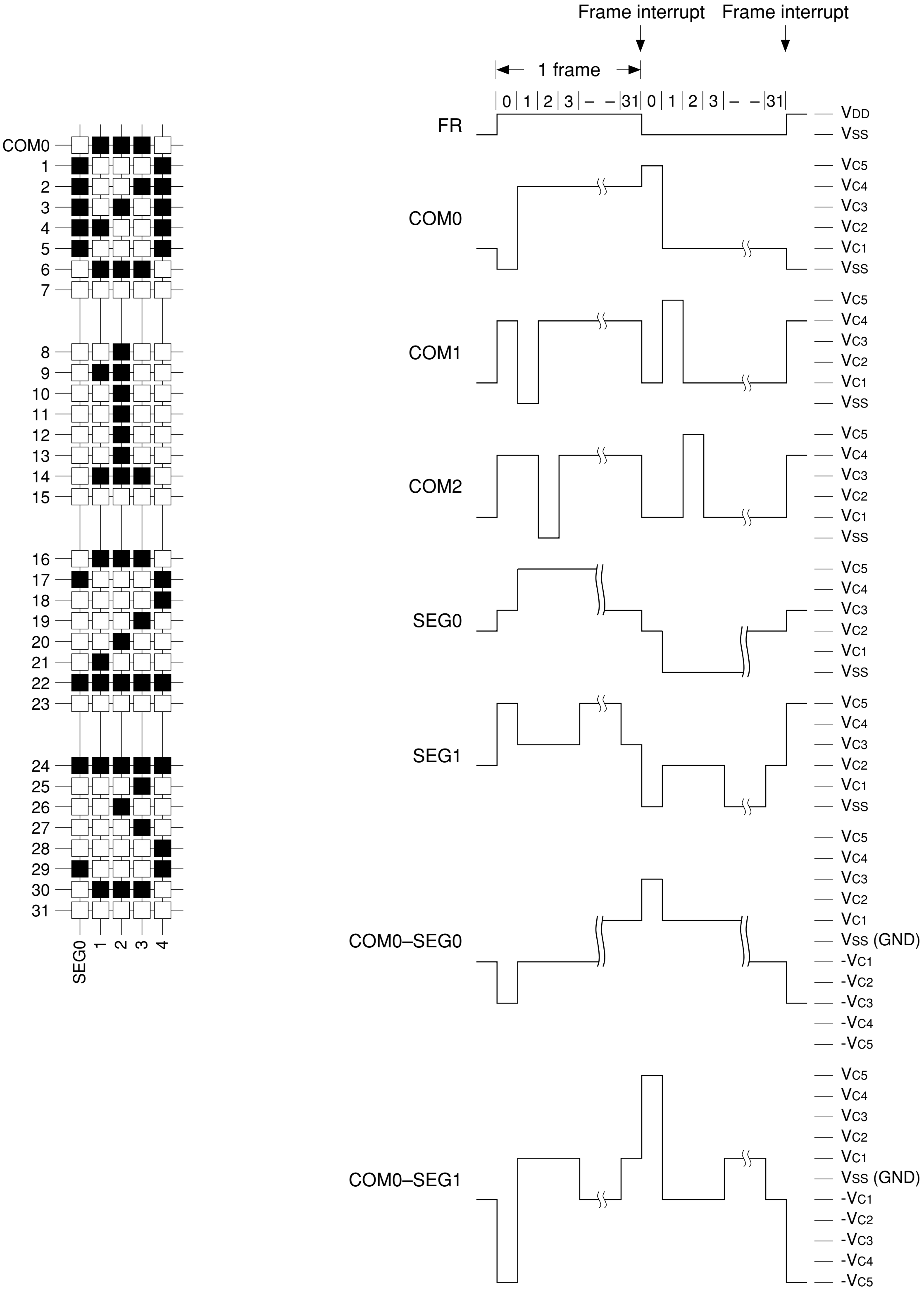


Figure 22.4.1 Drive Waveform for 1/32 Duty

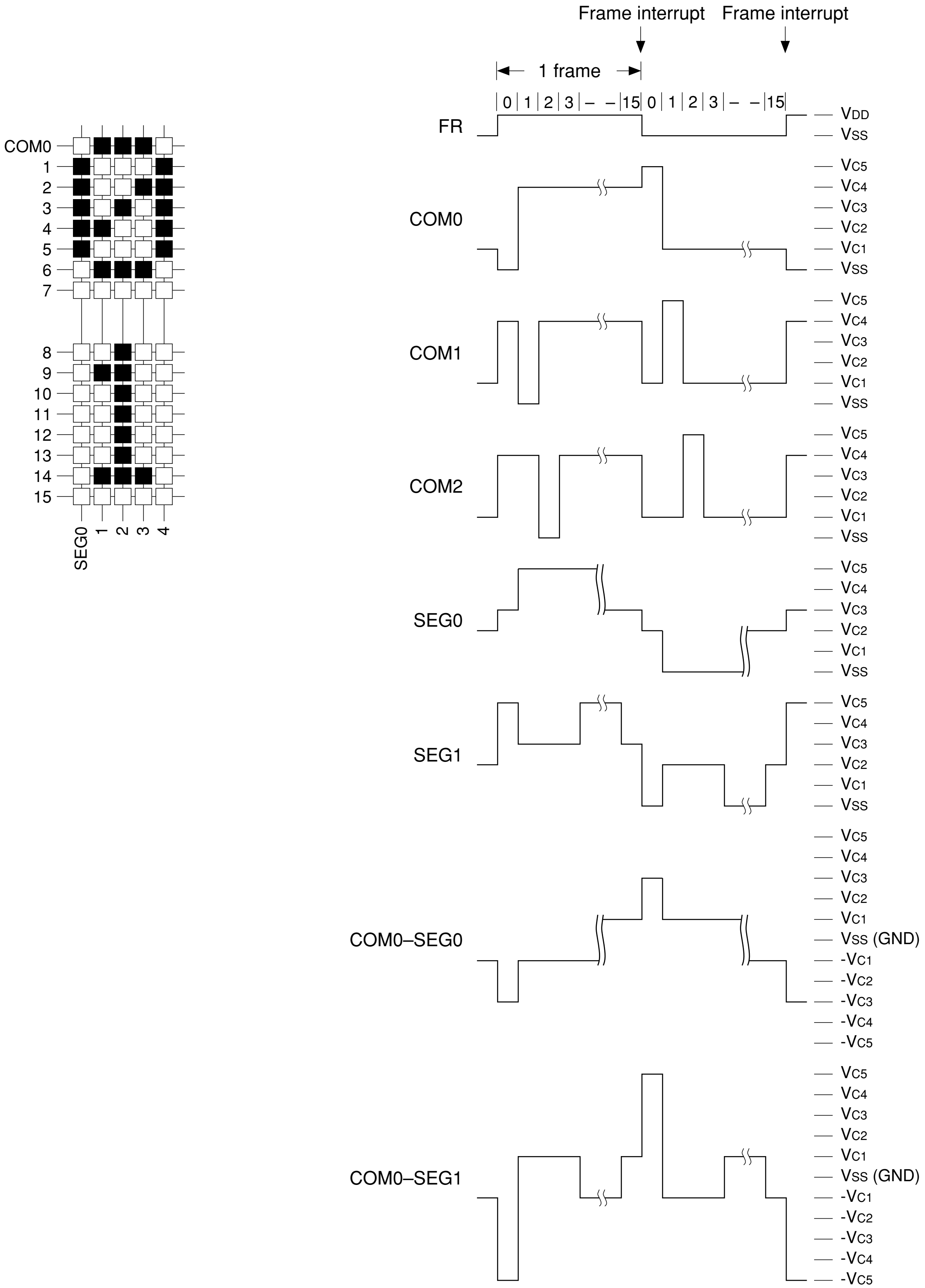


Figure 22.4.2 Drive Waveform for 1/16 Duty

22.5 Display Memory

The S1C17701 has a built-in 576-byte display memory located from address 0x80000 to address 0x8055f. The memory bit allocation to the COM/SEG pins will change according to the conditions below.

(1) Drive duty (1/32 or 1/16)
(2) SEG pin assignments (normal or reverse)
(3) COM pin assignments (normal or reverse)

Figures 22.5.1 and 22.5.2 show the correspondence between the display memory bits and the COM/SEG pins for each duty selection.

When a display memory bit is set to 1, the corresponding pixel on the LCD panel goes on; when it is set to 0, the pixel goes off. As it is a RAM, the display memory allows bit-unit control using logical operation (read-modify-write) instructions.

In the 576-byte display memory area, addresses unused for display can be used as general-purpose read/write memory.

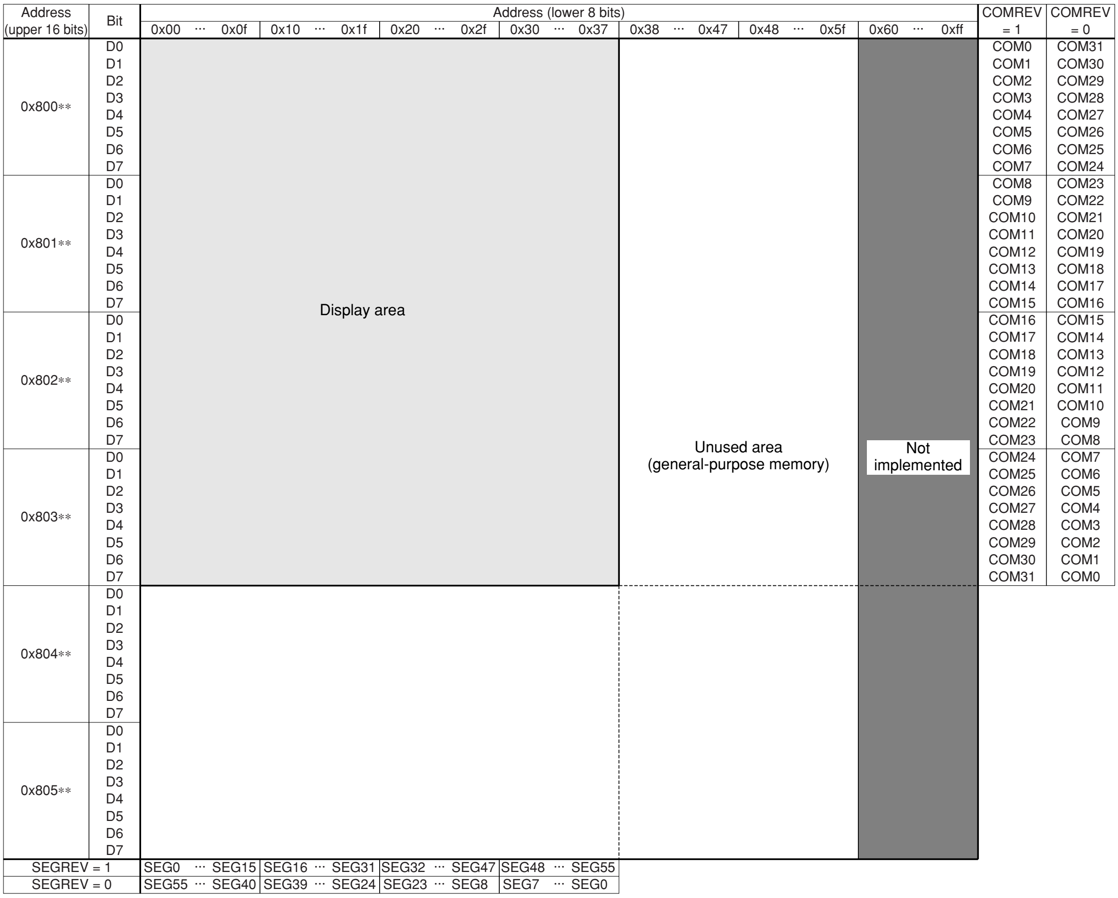

Address (upper 16 bits)
Bit
Address (lower 8 bits)
0x00 ··· 0x0f
0x10 ··· 0x1f
0x20 ··· 0x2f
0x30 ··· 0x37
0x38 ··· 0x47
0x48 ··· 0x5f
0x60 ··· 0xff
COMREV = 1
COMREV = 0
0x800**
0x801**
0x802**
0x803**
0x804**
0x805**
D0 D1 D2 D3 D4 D5 D6 D7
Display area
Unused area (general-purpose memory)
Not implemented
COM0 COM1 COM2 COM3 COM4 COM5 COM6 COM7 COM8 COM9 COM10 COM11 COM12 COM13 COM14 COM15 COM16 COM17 COM18 COM19 COM20 COM21 COM22 COM23 COM24 COM25 COM26 COM27 COM28 COM29 COM30 COM31
COM31 COM30 COM29 COM28 COM27 COM26 COM25 COM24 COM23 COM22 COM21 COM20 COM19 COM18 COM17 COM16 COM15 COM14 COM13 COM12 COM11 COM10 COM9 COM8 COM7 COM6 COM5 COM4 COM3 COM2 COM1 COM0
SEGREV = 1
SEG0 ··· SEG15
SEG16 ··· SEG31
SEG32 ··· SEG47
SEG48 ··· SEG55
SEGREV = 0
SEG55 ··· SEG40
SEG39 ··· SEG24
SEG23 ··· SEG8
SEG7 ··· SEG0

Figure 22.5.1 Display Memory Map (for 1/32 duty)

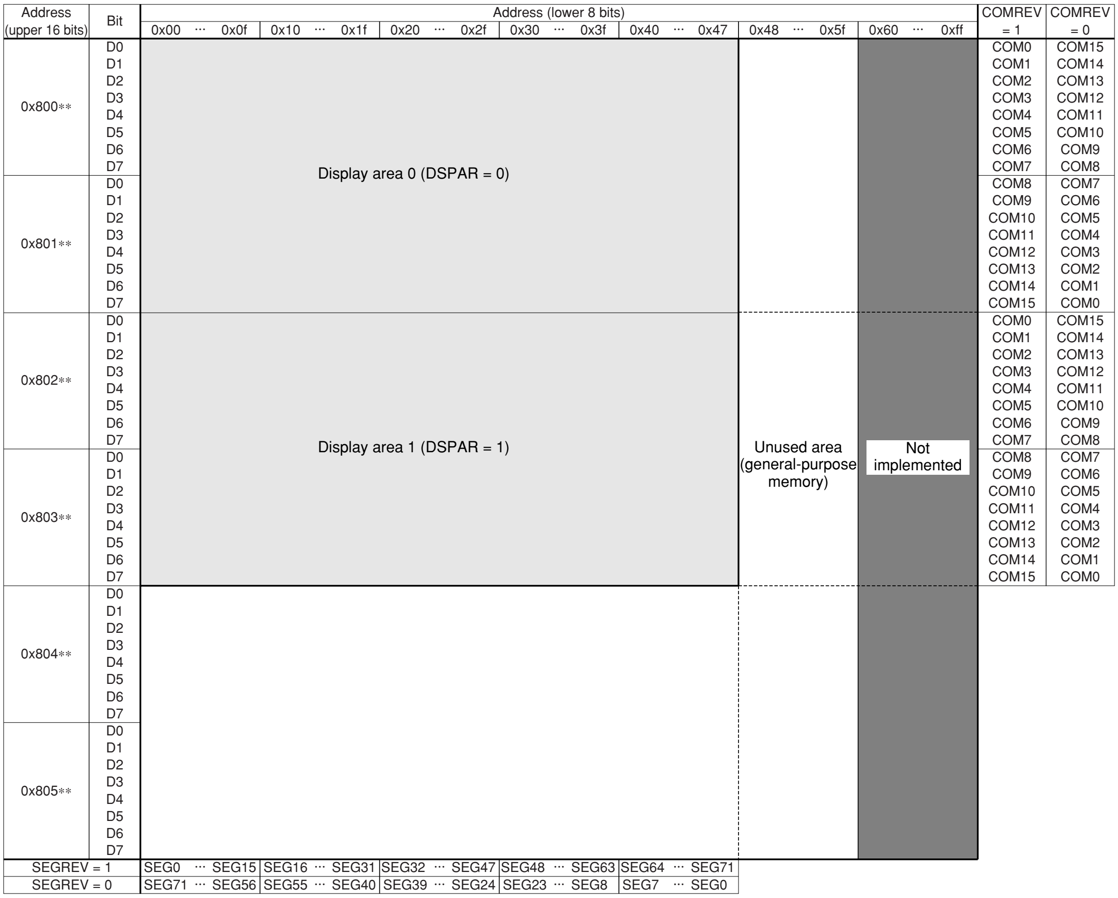


Figure 22.5.2 Display Memory Map (for 1/16 duty)

Selecting a display area (when 1/16 duty is selected)

When 1/16 duty is selected for the drive duty, two screen areas are reserved in the display memory and the area to be displayed can be selected using DSPAR (D5/LCD_DCTL register). When DSPAR is set to 0, display area 0 is selected; when it is set to 1, display area 1 is selected.

* **DSPAR**: Display Memory Area Control Bit in the LCD Display Control (LCD_DCTL) Register (D5/0x50a0)

SEG pin assignments

The memory allocation for the SEG pins can be reversed using SEGREV (D7/LCD_DCTL register). When SEGREV is 1 (default), the display memory addresses are allocated to the SEG pins in ascending order; when SEGREV is set to 0, the addresses are allocated in descending order. (See Figures 22.5.1 and 22.5.2.)

* **SEGREV**: Segment Output Assignment Control Bit in the LCD Display Control (LCD_DCTL) Register (D7/0x50a0)

COM pin assignments

The memory allocation for the COM pins can be reversed using COMREV (D6/LCD_DCTL register). When COMREV is 1 (default), the display memory bits are allocated to the COM pins in ascending order; when COMREV is set to 0, the addresses are allocated in descending order. (See Figures 22.5.1 and 22.5.2.)

* **COMREV**: Common Output Assignment Control Bit in the LCD Display Control (LCD_DCTL) Register (D6/0x50a0)

22.6 Display Control

22.6.1 Turning Display On and Off

The LCD display status can be controlled using DSPC[1:0] (D[1:0]/LCD_DCTL register).

∗ **DSPC[1:0]**: LCD Display Control Bits in the LCD Display Control (LCD_DCTL) Register (D[1:0]/0x50a0)

Table 22.6.1.1 LCD Display Control

DSPC[1:0]	LCD display
0x3	All off (static)
0x2	All on (dynamic)
0x1	Normal display
0x0	Display off

(Default: 0x0)

Set DSPC[1:0] to 0x1 to display normally. However, the clock must be supplied in advance (see Section 22.3).

When "Display off" is selected, the drive voltage supply from the LCD system voltage regulator is disabled and all the V_{C1} to V_{C5} pins go to V_{SS} level.

All on/off display is achieved by directly changing the LCD drive waveform to on or off and it does not affect the display memory data. The COM pins output a dynamic drive waveform when "All on" is selected or output a static drive waveform when "All off" is selected. This facility allows the program to blink screen without altering the display memory data.

DSPC[1:0] is reset to 0x0 (display off) after initial resetting.
DSPC[1:0] is not reset to 0x0 (display off) on execution of a slp command. DSPC[1:0] should be reset to 0x0 (display off) via software before executing a slp command, since switching to SLEEP mode with the LCD display left on will degrade the LCD.

22.6.2 LCD Contrast Adjustment

The LCD contrast can be adjusted in 16 steps. This facility is achieved to control the V_{C1} to V_{C5} voltages output from the LCD system voltage regulator.

∗ **LC[3:0]**: LCD Contrast Adjustment Bits in the LCD Contrast Adjust (LCD_CADJ) Register (D[3:0]/0x50a1)

Table 22.6.2.1 LCD Contrast Adjustment

LC[3:0]	Contrast
0xf	High (dark)
0xe	↑
:	:
0x1	↓
0x0	Low (light)

(Default: 0x0)

At initial reset, LC[3:0] is set to 0x0. Initialize LC[3:0] with software to set the display to the desired contrast.

22.6.3 Reverse Display

The display can be reversed (turns black pixels white and vice versa) simply by manipulating a control bit without altering the display memory data. Setting DSPREV (D4/LCD_DCTL register) to 0 reverses the display, and setting it to 1 returns the display to normal.

∗ **DSPREV**: Reverse Display Control Bit in the LCD Display Control (LCD_DCTL) Register (D4/0x50a0)

However, the display cannot be reversed when "All off" is selected with DSPC[1:0] (D[1:0]/LCD_DCTL register). The display can be reversed when "All on" is selected.

22.6.4 Controlling Gray Scale Display

The LCD display is capable of generating an interrupt in every frame. By controlling the pixels on and off using this interrupt, gray scale display can be realized.

Gray levels that can be produced depend on the LCD panel characteristics. The gray scale display should be performed by controlling the frame frequency and the frame cycles to turn the pixel on and off according to the panel characteristics.

See Section 22.7 for the frame interrupt.

22.7 LCD Interrupt

The LCD module can generate an interrupt by the frame signal.

Frame interrupt

This interrupt request occurs in every frame, and it sets the interrupt flag FRMIF (D0/LCD_IFLG register) in the LCD module to 1.
See Figures 22.4.1 and 22.4.2 for the interrupt timings.

* **FRMIF**: Frame Signal Interrupt Flag in the LCD Interrupt Flag (LCD_IFLG) Register (D0/0x50a6)

Set the FRMIE (D0/LCD_IMSK register) to 1 when using this interrupt. If FRMIE is set to 0 (default), an interrupt request by this cause will not be sent to the interrupt controller (ITC).

* **FRMIE**: Frame Signal Interrupt Enable Bit in the LCD Interrupt Mask (LCD_IMSK) Register (D0/0x50a5)

If FRMIF is set to 1 when the FRMIE has been set to 1 (interrupt enabled), the LCD module outputs the interrupt request signal to the ITC. The interrupt request signal sets the LCD interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The interrupt handler routine must reset (write 1 to) FRMIF in the LCD module, not the LCD interrupt flag in the ITC, to clear the cause of interrupt.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the FRMIF flag before the LCD interrupt is enabled using FRMIE.

ITC registers for LCD interrupt

According to the interrupt condition settings shown above, the LCD asserts the interrupt signal sent to the ITC at the rising edge of the frame signal. To generate an LCD interrupt, set the interrupt level and enable the interrupt using the ITC registers. The following shows the control bits for the LCD interrupt in the ITC.

Interrupt flag in the ITC
* **EIFT6**: LCD Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D6/0x4300)

Interrupt enable bit in the ITC
* **EIEN6**: LCD Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D6/0x4302)

Interrupt level setup bits in the ITC
* **EILV6[2:0]**: LCD Interrupt Level Bits in the External Interrupt Level Setup (ITC_ELV3) Register 3 (D[2:0]/0x430c)

Interrupt trigger mode select bit in the ITC (fixed at 1)
* **EITG6**: LCD Interrupt Trigger Mode Select Bit in the External Interrupt Level Setup (ITC_ELV3) Register 3 (D4/0x430c)

The LCD interrupt signal sets EIFT6 to 1. If EIEN6 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the LCD interrupt, set EIEN6 to 0. EIFT6 is always set to 1 by the interrupt signal sent from the LCD module, regardless of how EIEN6 is set (even when set to 0).
EILV6[2:0] sets the interrupt level (0 to 7) of the LCD interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The LCD interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Chapter 6, “Interrupt Controller (ITC).”

Note: The settings shown below are required to manage the cause-of-interrupt occurrence status using the interrupt flag in the LCD module.

1. Set the LCD interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the LCD interrupt flag FRMIF of the LCD module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

Interrupt vector

The following shows the vector number and vector address for the LCD interrupt:

Vector number: 10 (0x0a)
Vector address: 0x8028

22.8 Details of Control Registers

Table 22.8.1 List of LCD Registers

Address	Register name		Function
0x50a0	LCD_DCTL	LCD Display Control Register	Controls the LCD display.
0x50a1	LCD_CADJ	LCD Contrast Adjust Register	Controls the contrast.
0x50a2	LCD_CCTL	LCD Clock Control Register	Controls the LCD clock duty.
0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	Controls the LCD drive voltage regulator.
0x50a4	LCD_PWR	LCD Power Voltage Booster Control Register	Controls the LCD voltage booster.
0x50a5	LCD_IMSK	LCD Interrupt Mask Register	Enables/disables interrupt.
0x50a6	LCD_IFLG	LCD Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The following describes each LCD register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x50a0: LCD Display Control Register (LCD_DCTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
LCD Display Control Register (LCD_DCTL)	**0x50a0** (8 bits)	D7	**SEGREV**	Segment output assignment control	1	Normal	0	Reverse	1	R/W	
		D6	**COMREV**	Common output assignment control	1	Normal	0	Reverse	1	R/W	
		D5	**DSPAR**	Display memory area control	1	Area 1	0	Area 0	0	R/W	
		D4	**DSPREV**	Reverse display control	1	Normal	0	Reverse	1	R/W	
		D3–2	**–**	reserved	–				–	–	0 when being read.
		D1–0	**DSPC[1:0]**	LCD display control	DSPC[1:0]		Display		0x0	R/W	
					0x3		All off				
					0x2		All on				
					0x1		Normal display				
					0x0		Display off				

D7 SEGREV: Segment Output Assignment Control Bit

Reverses the memory allocation for the SEG terminals.

1 (R/W): Normal (default)

0 (R/W): Reverse

When SEGREV is 1 (default), the display memory addresses are allocated to the SEG pins in ascending order; when SEGREV is set to 0, the addresses are allocated in descending order. (See Figures 22.5.1 and 22.5.2.)

D6 COMREV: Common Output Assignment Control Bit

Reverses the memory bit allocation for the COM terminals.

1 (R/W): Normal (default)

0 (R/W): Reverse

When COMREV is 1 (default), the display memory bits are allocated to the COM pins in ascending order; when COMREV is set to 0, the addresses are allocated in descending order. (See Figures 22.5.1 and 22.5.2.)

D5 DSPAR: Display Memory Area Control Bit

Selects the display area for 1/16 duty drive.

1 (R/W): Display area 1

0 (R/W): Display area 0 (default)

When 1/16 duty is selected for the drive duty, two screen areas are reserved in the display memory and the area to be displayed can be selected using DSPAR. When DSPAR is set to 0, display area 0 is selected; when it is set to 1, display area 1 is selected. For the display area, see Figure 22.5.2.

D4 DSPREV: Reverse Display Control Bit

Reverses the display (negative display) on the LCD.

1 (R/W): Normal display (default)

0 (R/W): Reverse display

Setting DSPREV to 0 reverses the display (turns black pixels white and vice versa), and setting it to 1 returns the display to normal. This operation does not affect the contents of the display memory.

D[3:2] Reserved

D[1:0] DSPC[1:0]: LCD Display Control Bit

Controls the display on the LCD.

Table 22.8.2 LCD Display Control

DSPC[1:0]	LCD display
0x3	All off (static)
0x2	All on (dynamic)
0x1	Normal display
0x0	Display off

(Default: 0x0)

Set DSPC[1:0] to 0x1 to display normally. However, the clock must be supplied in advance (see Section 22.3).

When "Display off" is selected, the drive voltage supply from the LCD system voltage regulator is disabled and all the V_{C1} to V_{C5} pins go to V_{SS} level.

All on/off display is achieved by directly changing the LCD drive waveform to on or off and it does not affect the display memory data. The COM pins output a dynamic drive waveform when "All on" is selected or output a static drive waveform when "All off" is selected. This facility allows the program to blink screen without altering the display memory data.

DSPC[1:0] is reset to 0x0 (display off) after initial resetting.
DSPC[1:0] is not reset to 0x0 (display off) on execution of a slp command. DSPC[1:0] should be reset to 0x0 (display off) via software before executing a slp command, since switching to SLEEP mode with the LCD display left on will degrade the LCD.

0x50a1: LCD Contrast Adjust Register (LCD_CADJ)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
LCD Contrast Adjust Register (LCD_CADJ)	**0x50a1** (8 bits)	D7–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**LC[3:0]**	LCD contrast adjustment	LC[3:0]	Display	0x0	R/W	
					0xf : 0x0	Dark : Light			

D[7:4] Reserved

D[3:0] LC[3:0]: LCD Contrast Adjustment Bits

Adjusts the LCD contrast. This facility is achieved to control the V_{C1} to V_{C5} voltages output from the LCD system voltage regulator.

Table 22.8.3 LCD Contrast Adjustment

LC[3:0]	Contrast
0xf	High (dark)
0xe	↑
:	:
0x1	↓
0x0	Low (light)

(Default: 0x0)

At initial reset, LC[3:0] is set to 0x0. Initialize LC[3:0] with software to set the display to the desired contrast.

0x50a2: LCD Clock Control Register (LCD_CCTL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
LCD Clock Control Register (LCD_CCTL)	**0x50a2** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**LDUTY[1:0]**	LCD duty select	LDUTY[1:0]	Duty	0x2	R/W	
					0x3	reserved			
					0x2	1/32			
					0x1	1/16			
					0x0	reserved			

D[7:2] Reserved

D[1:0] LDUTY[1:0]: LCD Duty Select Bits

Selects a drive duty.

Table 22.8.4 Setting the Drive Duty

LDUTY[1:0]	Duty	Effective COM pins	Effective SEG pins	Max number of pixels
0x3	Reserved	–	–	–
0x2	1/32	COM0–COM31	SEG0–SEG55	1,792 pixels
0x1	1/16	COM0–COM15	SEG0–SEG71	1,152 pixels
0x0	Reserved	–	–	–

(Default: 0x2)

0x50a3: LCD Voltage Regulator Control Register (LCD_VREG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
LCD Voltage Regulator Control Register (LCD_VREG)	**0x50a3** (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**LHVLD**	LCD heavy load protection mode	1	On	0	Off	0	R/W	
		D3–0	**–**	reserved	–				–	–	0 when being read.

For details of the control bits, see “0x50a3: LCD Voltage Regulator Control Register (LCD_VREG)” in Section 4.5.

0x50a4: LCD Power Voltage Booster Control Register (LCD_PWR)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
LCD Power Voltage Booster Control Register (LCD_PWR)	**0x50a4** (8 bits)	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**VDSEL**	Regulator power source select	1	V_{D2}	0	V_{DD}	0	R/W	
		D0	**PBON**	Power voltage booster control	1	On	0	Off	0	R/W	

For details of the control bits, see “0x50a4: LCD Power Voltage Booster Control Register (LCD_PWR)” in Section 4.5.

0x50a5: LCD Interrupt Mask Register (LCD_IMSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Interrupt Mask Register (LCD_IMSK)	**0x50a5** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**FRMIE**	Frame signal interrupt enable	1 Enable 0 Disable	0	R/W	

D[7:1] Reserved

D0 FRMIE: Frame Signal Interrupt Enable Bit

Enables/disables the frame interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

Setting FRMIE to 1 enables the LCD module to request interrupts to the ITC; setting to 0 disables the interrupt.

In addition, it is necessary to set the LCD interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

0x50a6: LCD Interrupt Flag Register (LCD_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
LCD Interrupt Flag Register (LCD_IFLG)	**0x50a6** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**FRMIF**	Frame signal interrupt flag	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.

D[7:1] Reserved

D0 FRMIF: Frame Signal Interrupt Flag

This is the interrupt flag to indicate the frame interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

FRMIF is the interrupt flag for the LCD module. The interrupt flag is set to 1 at the rising edge of the frame signal. If FRMIE (D0/LCD_IMSK register) has been set to 1 at this time, the LCD interrupt request signal is output to the ITC. The interrupt request signal sets the LCD interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

The settings shown below are required to manage the cause-of-interrupt occurrence status using this register.

1. Set the LCD interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the LCD interrupt flag of the LCD module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

The FRMIF flag is reset by writing 1.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the FRMIF flag before the LCD interrupt is enabled using FRMIE.

22.9 Precautions

- To avoid occurrence of unnecessary interrupts, be sure to reset FRMIF (D0/LCD_IFLG register) before the LCD interrupt is enabled using FRMIE (D0/LCD_IMSK register).
- For precautions on the LCD power supply, see Section 4.6, “Precautions.”

23 Supply Voltage Detector (SVD)

23.1 Outline of the SVD module

The S1C17701 is equipped with a supply voltage detector (SVD) to detect supply voltage drop. The SVD module allows software to turn the circuit on and off, to set an evaluate voltage level and to read the detection results. Also it can generate an interrupt when voltage drop has been detected.
Figure 23.1.1 shows the structure of the SVD module.

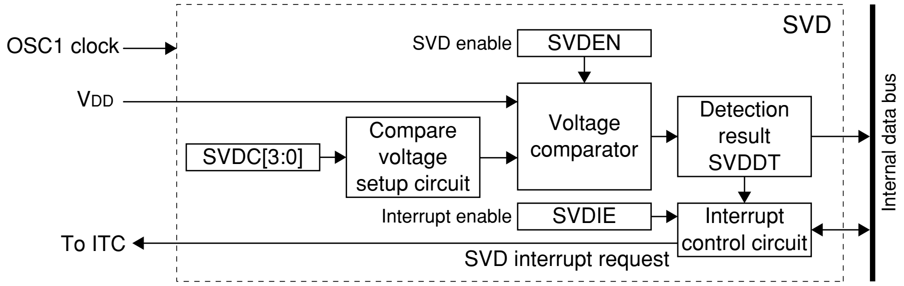


Figure 23.1.1 Structure of SVD Module

23.2 Setting a Compare Voltage

The SVD module compares the voltage set with software and the supply voltage (V_{DD}) and outputs the results to indicate whether or not the supply voltage is equal to or higher than the compare voltage. The compare voltage can be selected from the 13 levels listed in Table 23.2.1 using SVDC[3:0] (D[3:0]/SVD_CMP register).

∗ **SVDC[3:0]**: SVD Compare Voltage Select Bits in the SVD Compare Voltage (SVD_CMP) Register (D[3:0]/0x5101)

Table 23.2.1 Setting the Compare Voltage

SVDC[3:0]	Compare voltage
0xf	2.7 V
0xe	2.6 V
0xd	2.5 V
0xc	2.4 V
0xb	2.3 V
0xa	2.2 V
0x9	2.1 V
0x8	2.05 V
0x7	2.0 V
0x6	1.95 V
0x5	1.9 V
0x4	1.85 V
0x3	1.8 V
0x2 to 0x0	Reserved

(Default: 0x0)

23.3 Controlling the SVD Operation

The SVD module starts operating to detect the supply voltage level by writing 1 to SVDEN (D0/SVD_EN register) and stops when 0 is written to SVDEN.

∗ **SVDEN**: SVD Enable Bit in the SVD Enable (SVD_EN) Register (D0/0x5100)

The detection results can be read from SVDDT (D0/SVD_RSLT register).

∗ **SVDDT**: SVD Detection Result Bit in the SVD Detection Result (SVD_RSLT) Register (D0/0x5102)

The read value indicates the detection result as follows:

- SVDDT = 0 when supply voltage (V_{DD}) ≥ compare voltage
- SVDDT = 1 when supply voltage (V_{DD}) < compare voltage

Furthermore, if SVDEN is set to 1 when the SVD interrupt has been enabled, an interrupt will occur when the supply voltage drops under the compare voltage and it sets the detection result to 1. By using this interrupt, the application program can display for battery exhaustion or setting heavy load protection mode. See the subsequent section for control of the interrupt.
When an interrupt occurs due to temporary voltage sags, it will not be canceled even if the voltage returns over the compare voltage. Read SVDDT in the interrupt handler routine to check the current status.

Notes:

- After the SVD module starts operating, 500 µs (max.) is required until a stable detection result can be read. When reading the detection results without using an interrupt, wait for the stabilization time before reading SVDDT (D0/SVD_RSLT register) after writing 1 to SVDEN (D0/SVD_EN register).
- The SVD operation increases current consumption. Therefore, set SVDEN to 0 to disable the SVD operation if supply voltage detection is not necessary.

23.4 SVD Interrupt

The SVD module can generate an interrupt when supply voltage drop has been detected.

Supply voltage drop detection interrupt

This interrupt request occurs when the SVD module being operated (SVDEN (D0/SVD_EN register) = 1) detects that the supply voltage (VDD) drops under the compare voltage, and it sets the interrupt flag SVDIF (D0/SVD_IFLG register) in the SVD module to 1. Note that the SVDIF, which has been set to 1, cannot be reset even if the supply voltage returns over the compare voltage after that.

* **SVDIF**: SVD Interrupt Flag in the SVD Interrupt Flag (SVD_IFLG) Register (D0/0x5104)

Set the SVDIE bit (D0/SVD_IMSK register) to 1 when using this interrupt. If SVDIE is set to 0 (default), an interrupt request by this cause will not be sent to the interrupt controller (ITC).

* **SVDIE**: SVD Interrupt Enable Bit in the SVD Interrupt Mask (SVD_IMSK) Register (D0/0x5103)

If SVDIF is set to 1 when the SVDIE has been set to 1 (interrupt enabled), the SVD module outputs the interrupt request signal to the ITC. The interrupt request signal sets the SVD interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The interrupt handler routine must reset (write 1 to) SVDIF in the SVD module, not the SVD interrupt flag in the ITC, to clear the cause of interrupt.

Notes:
- No interrupt will occur if the supply voltage is already lower than the compare voltage at the time SVDIE is set to 1 (interrupt enabled).
- To avoid occurrence of unnecessary interrupts, be sure to reset the SVDIF flag before the SVD interrupt is enabled using SVDIE.

ITC registers for SVD interrupt

When the SVD module has detected supply voltage drop, the SVD asserts the interrupt signal sent to the ITC according to the interrupt condition settings shown above.
To generate an SVD interrupt, set the interrupt level and enable the interrupt using the ITC registers.
The following shows the control bits for the SVD interrupt in the ITC.

Interrupt flag in the ITC

* **EIFT5**: SVD Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D5/0x4300)

Interrupt enable bit in the ITC

* **EIEN5**: SVD Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D5/0x4302)

Interrupt level setup bits in the ITC

* **EILV5[2:0]**: SVD Interrupt Level Bits in the External Interrupt Level Setup (ITC_ELV2) Register 2 (D[10:8]/0x430a)

Interrupt trigger mode select bit in the ITC (fixed at 1)

* **EITG5**: SVD Interrupt Trigger Mode Select Bit in the External Interrupt Level Setup (ITC_ELV2) Register 2 (D12/0x430a)

When the SVD module has detected supply voltage drop, EIFT5 is set to 1. If EIEN5 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the SVD interrupt, set EIEN5 to 0. EIFT5 is always set to 1 by the interrupt signal sent from the SVD module, regardless of how EIEN5 is set (even when set to 0).
EILV5[2:0] sets the interrupt level (0 to 7) of the SVD interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The SVD interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Chapter 6, “Interrupt Controller (ITC).”

Note: The settings shown below are required to manage the cause-of-interrupt occurrence status using the interrupt flag in the SVD module.

1. Set the SVD interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the SVDIF interrupt flag of the SVD module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

Interrupt vector

The following shows the vector number and vector address for the SVD interrupt:

Vector number: 9 (0x09)
Vector address: 0x8024

23.5 Details of Control Registers

Table 23.5.1 List of SVD Registers

Address	Register name		Function
0x5100	SVD_EN	SVD Enable Register	Enables/disables the SVD operation.
0x5101	SVD_CMP	SVD Compare Voltage Register	Sets compare voltage.
0x5102	SVD_RSLT	SVD Detection Result Register	Voltage detection results
0x5103	SVD_IMSK	SVD Interrupt Mask Register	Enables/disables interrupt.
0x5104	SVD_IFLG	SVD Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The following describes each SVD register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5100: SVD Enable Register (SVD_EN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SVD Enable Register (SVD_EN)	**0x5100** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**SVDEN**	SVD enable	1	Enable	0	Disable	0	R/W	

D[7:1] Reserved

D0 SVDEN: SVD Enable Bit

Enables or disables the SVD module to operate.

1 (R/W): Enable

0 (R/W): Disable (default)

The SVD module starts operating to detect the supply voltage level by setting SVDEN to 1 and stops when it is set to 0.

Notes:
- After the SVD module starts operating, 500 µs (max.) is required until a stable detection result can be read. When reading the detection results without using an interrupt, wait for the stabilization time before reading SVDDT (D0/SVD_RSLT register) after writing 1 to SVDEN.
- The SVD operation increases current consumption. Therefore, set SVDEN to 0 to disable the SVD operation if supply voltage detection is not necessary.

0x5101: SVD Compare Voltage Register (SVD_CMP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
SVD Compare Voltage Register (SVD_CMP)	**0x5101** (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–0	**SVDC[3:0]**	SVD compare voltage	SVDC[3:0]	Voltage	0x0	R/W	
					0xf	2.7 V			
					0xe	2.6 V			
					0xd	2.5 V			
					0xc	2.4 V			
					0xb	2.3 V			
					0xa	2.2 V			
					0x9	2.1 V			
					0x8	2.05 V			
					0x7	2.0 V			
					0x6	1.95 V			
					0x5	1.9 V			
					0x4	1.85 V			
					0x3	1.8 V			
					0x2	–			
					0x1	–			
					0x0	–			

D[7:4] Reserved

D[3:0] SVDC[3:0]: SVD Compare Voltage Select Bits

Selects a compare voltage for detecting supply voltage drop from the 13 levels.

Table 23.5.2 Setting the Compare Voltage

SVDC[3:0]	Compare voltage
0xf	2.7 V
0xe	2.6 V
0xd	2.5 V
0xc	2.4 V
0xb	2.3 V
0xa	2.2 V
0x9	2.1 V
0x8	2.05 V
0x7	2.0 V
0x6	1.95 V
0x5	1.9 V
0x4	1.85 V
0x3	1.8 V
0x2 to 0x0	Reserved

(Default: 0x0)

The SVD module compares the voltage set with SVDC[3:0] and the supply voltage (V_{DD}) and output the results to indicate whether or not the supply voltage is equal to or higher than the compare voltage.

0x5102: SVD Detection Result Register (SVD_RSLT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SVD Detection Result Register (SVD_RSLT)	**0x5102** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**SVDDT**	SVD detection result	1	Low	0	Normal	×	R	

D[7:1] Reserved

D0 SVDDT: SVD Detection Result Bit

Indicates the supply voltage detection results.

1 (R): Supply voltage (VDD) < Compare voltage

0 (R): Supply voltage (VDD) ≥ Compare voltage

The SVD module keeps comparing the supply voltage (VDD) and the voltage level set with SVDC[3:0] (D[3:0]/SVD_CMP register) while SVDEN (D0/SVD_EN register) is set to 1. By reading SVDDT, the current supply voltage status can be monitored.

0x5103: SVD Interrupt Mask Register (SVD_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SVD Interrupt Mask Register (SVD_IMSK)	**0x5103** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**SVDIE**	SVD interrupt enable	1	Enable	0	Disable	0	R/W	

D[7:1] Reserved

D0 SVDIE: SVD Interrupt Enable Bit

Enables/disables the supply voltage drop interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

Setting SVDIE to 1 enables the SVD module to request interrupts to the ITC; setting to 0 disables the interrupt.

In addition, it is necessary to set the SVD interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

0x5104: SVD Interrupt Flag Register (SVD_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SVD Interrupt Flag Register (SVD_IFLG)	**0x5104** (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	**SVDIF**	SVD interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

D[7:1] Reserved

D0 SVDIF: SVD Interrupt Flag

This is the interrupt flag to indicate the supply voltage drop interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Has no effect

SVDIF is the interrupt flag for the SVD module. The interrupt flag is set to 1 when the SVD module has detected supply voltage drop. If SVDIE (D0/SVD_IMSK register) has been set to 1 at this time, the SVD interrupt request signal is output to the ITC. The interrupt request signal sets the SVD interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

The settings shown below are required to manage the cause-of-interrupt occurrence status using this register.

1. Set the SVD interrupt trigger mode in the ITC to level trigger.
2. After an interrupt occurs, reset the SVDIF interrupt flag of the SVD module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

The SVDIF flag is reset by writing 1.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the SVDIF flag before the SVD interrupt is enabled using SVDIE (D0/SVD_IMSK register).

23.6 Precautions

- After the SVD module starts operating, 500 µs (max.) is required until a stable detection result can be read. When reading the detection results without using an interrupt, wait for the stabilization time before reading SVDDT (D0/SVD_RSLT register) after writing 1 to SVDEN (D0/SVD_EN register).
- The SVD operation increases current consumption. Therefore, set SVDEN (D0/SVD_EN register) to 0 to disable the SVD operation if supply voltage detection is not necessary.
- To avoid occurrence of unnecessary interrupts, be sure to reset the SVDIF flag (D0/SVD_IFLG register) before the SVD interrupt is enabled using SVDIE (D0/SVD_IMSK register).

24 On-chip Debugger (DBG)

24.1 Resource Requirements and Debugging Tools

Work area for debugging

A 64-byte work area is required for debugging. In the S1C17701, the address range from 0x000fc0 to 0x000fff in the RAM is reserved as the work area for debugging. When using the debug functions, do not access this area from the application program.

The debug RAM start address can be read out from the DBRAM register (0xffff90).

Debugging tools

Debugging is performed by connecting the ICD (In-Circuit Debugger) such as S5U1C17001H (ICD Mini) to the debug pins of the S1C17701 and entering debug commands from the debugger being run on a personal computer. The tools listed below are required for debugging.

- S1C17 Family In-Circuit Debugger (e.g. S5U1C17001H)
- S1C17 Family C Compiler Package (e.g. S5U1C17001C)

Debug pins

The ICD (e.g. S5U1C17001H) is connected to the debug pins listed below.

Table 24.1.1 List of Debug Pins

Pin name	I/O	Size	Function
DCLK (P31)	O	1	On-chip debugger clock output pin This pin outputs a clock to the ICD Mini (S5U1C17001H).
DSIO (P33)	I/O	1	On-chip debugger data input/output pin This pin inputs/outputs data for debugging and inputs a break signal.
DST2 (P32)	O	1	On-chip debugger status signal output pin This pin outputs the processor status during debugging.

The on-chip debugger input/output pins (DCLK, DST2, DSIO) are shared with the I/O ports (P31, P32, P33) and they are initialized as debug pins by default. When the debug function is not used, these pins can be configured for general-purpose I/O ports using the P3_PMUX register. Set the control bits shown below to 1 to configure the pins for the I/O ports.

DCLK → P31

∗ **P31MUX**: P31 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D1/0x52a3)

DST2 → P32

∗ **P32MUX**: P32 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D2/0x52a3)

DSIO → P33

∗ **P33MUX**: P33 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D3/0x52a3)

For details on switching pin function, see Section 10.2, “Selecting I/O Pin Functions (Port MUX).”

24.2 Operating Status after Debugging Break Occurs

When the `brk` instruction is executed or a break signal (low) is input to the DSIO pin, a debug interrupt occurs and the S1C17 Core enters debug mode. This status continues until the `retd` instruction is executed.
In debug mode, hardware interrupts and NMI cannot be accepted.
By default, the peripheral modules are stopped in debug mode. This status can be changed so that they will operate during debugging.
The LCD driver continues in operating status when a debug interrupt occurs during debugging.

Peripheral modules that operate with a prescaler output clock

- 8-bit timer
- 16-bit timers
- PWM & capture timer
- Remote controller
- P ports
- UART
- SPI
- I^{2}C

By default, the prescaler stops in debug mode. Therefore, the peripheral modules listed above are also stopped as they use a prescaler output clock. The prescaler provides PRUND (D1/PSC_CTL register) for specifying prescaler operating condition in debug mode. When PRUND is set to 1, the prescaler operates in debug mode. This allows the above peripheral modules to operate. When PRUND is 0 (default), the prescaler and above peripheral modules stop operating when the S1C17 Core enters debug mode.

∗ **PRUND**: Prescaler Run/Stop Setting (in Debug Mode) Bit in the Prescaler Control (PSC_CTL) Register (D1/0x4020)

Peripheral modules that operate with the OSC1 clock

- Clock timer
- Watchdog timer
- Stopwatch timer

The MISC register provides O1DBG (D0/MISC_OSC1 register) to specify the operating condition for the OSC1 peripheral modules (listed above) in debug mode. When O1DBG is set to 1, the OSC1 peripheral modules operate in debug mode. When O1DBG is 0 (default), the OSC1 peripheral modules stop operating when the S1C17 Core enters debug mode.

∗ **O1DBG**: OSC1 Peripheral Control (in Debug Mode) Bit in the OSC1 Peripheral Control (MISC_OSC1) Register (D0/0x5322)

Note that the 8-bit OSC1 timer does not stop in debug mode even if O1DBG is set to 1.

24.3 Details of Control Registers

Table 24.3.1 List of Registers for Debugging

Address	Register name		Function
0x5322	MISC_OSC1	OSC1 Peripheral Control Register	Selects the OSC1 peripheral operation in debug mode.
0xffff90	DBRAM	Debug RAM Base Register	Indicates the debug RAM base address.

The following describes the registers for debugging individually.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5322: OSC1 Peripheral Control Register (MISC_OSC1)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
OSC1 Peripheral Control Register (MISC_OSC1)	**0x5322** (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	**O1DBG**	OSC1 peripheral control in debug mode	1	Run	0	Stop	0	R/W	

D[7:1] Reserved

D0 O1DBG: OSC1 Peripheral Control in Debug Mode Bit

Sets the operating condition for the OSC1 peripheral modules in debug mode.

1 (R/W): Operate

0 (R/W): Stop (default)

The following lists the OSC1 peripheral modules that operate with the OSC1 clock:

- Clock timer
- Watchdog timer
- Stopwatch timer

Note that the 8-bit OSC1 timer does not stop in debug mode even if O1DBG is set to 1.

0xffff90: Debug RAM Base Register (DBRAM)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug RAM Base Register (DBRAM)	**0xffff90** (32 bits)	D31–24	–	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	**DBRAM[23:0]**	Debug RAM base address	0xfc0	0xfc0	R	

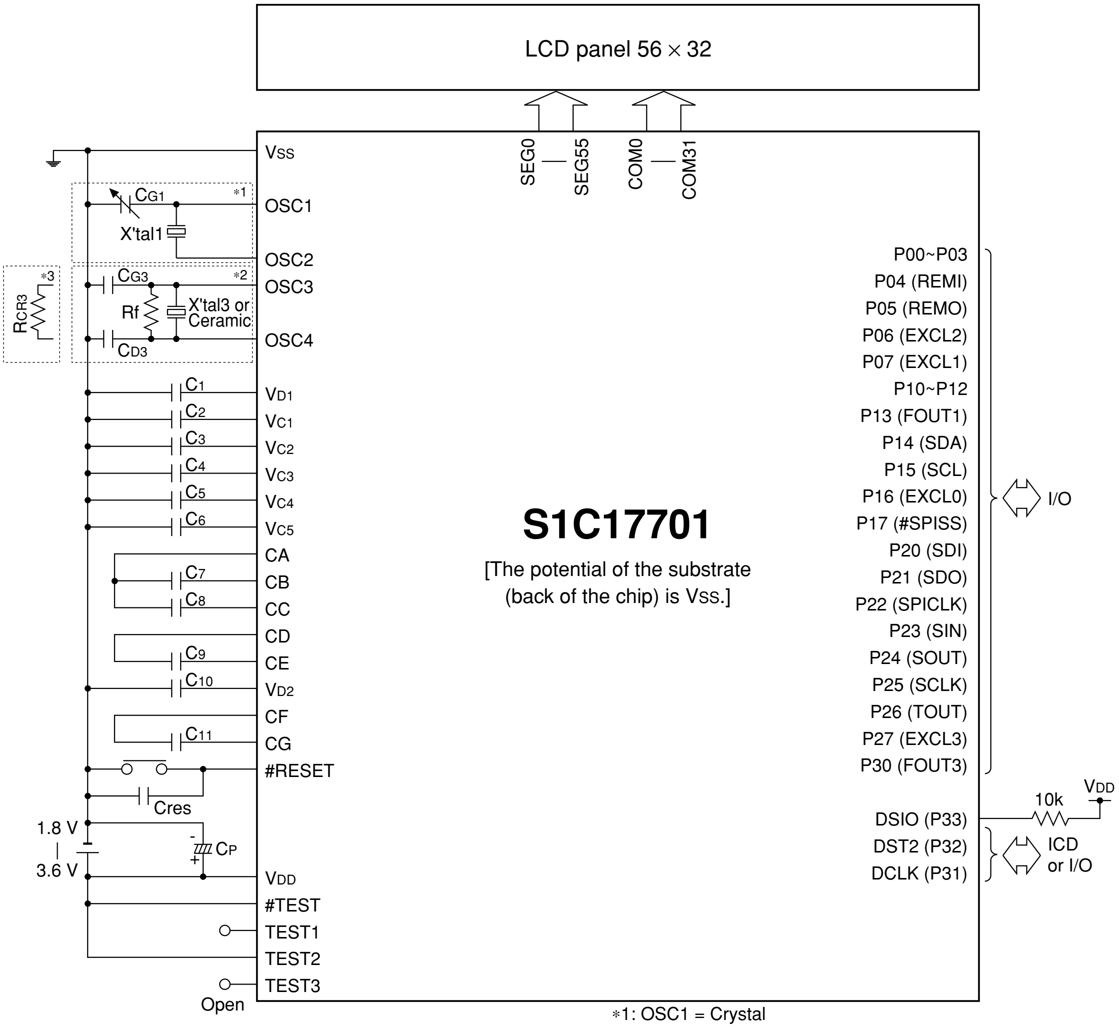
D[31:24] Unused (fixed at 0)

D[23:0] **DBRAM[23:0]: Debug RAM Base Address Bits**

This is a read-only register that contains the start address of a work area (64 bytes) for debugging.

THIS PAGE IS BLANK.

25 Basic External Wiring Diagram



∗1: OSC1 = Crystal
∗2: OSC3 = Crystal or Ceramic (S1C17701F00B100)
∗3: OSC3 = CR (S1C17701F00E100)

Recommended values for external parts

Symbol	Name	Recommended value
X'tal1	Crystal oscillator	32.768 kHz
C_{G1}	Trimmer capacitor	0–25 pF
X'tal3	Crystal oscillator	0.2–8 MHz
Ceramic	Ceramic oscillator	0.2–8 MHz
Rf	Feedback resistor	1 MΩ
C_{G3}	Gate capacitor	15 pF (crystal) 30 pF (ceramic)
C_{D3}	Drain capacitor	15 pF (crystal) 30 pF (ceramic)
R_{CR3}	Resistor for CR oscillation	30 kΩ

Symbol	Name	Recommended value
C_1	Capacitor between V_{SS} and V_{D1}	0.1 µF
C_2	Capacitor between V_{SS} and V_{C1}	0.1 µF
C_3	Capacitor between V_{SS} and V_{C2}	0.1 µF
C_4	Capacitor between V_{SS} and V_{C3}	0.1 µF
C_5	Capacitor between V_{SS} and V_{C4}	0.1 µF
C_6	Capacitor between V_{SS} and V_{C5}	0.1 µF
C_7–C_9	Booster capacitors	0.1 µF
C_{10}	Capacitor between V_{SS} and V_{D2}	0.1 µF
C_{11}	Booster capacitor	0.1 µF
C_P	Capacitor for power supply	3.3 µF
Cres	Capacitor for #RESET terminal	0.47 µF

THIS PAGE IS BLANK.

26 Electrical Characteristics

26.1 Absolute Maximum Rating

(V_{SS} = 0V)

Item	Symbol	Condition	Rated value	Unit
Power supply voltage	V_{DD}		-0.3 to 4.0	V
LCD power voltage	V_{C5}		-0.3 to 6.0	V
Input voltage	V_I		-0.3 to V_{DD} + 0.3	V
Output voltage	V_O		-0.3 to V_{DD} + 0.3	V
High level output current	I_{OH}	1 pin	-5	mA
		Total of all pins	-20	mA
Low level output current	I_{OL}	1 pin	5	mA
		Total of all pins	20	mA
Permitted loss *1	V_O		200	mW
Operating temperature	Ta		-20 to 70	°C
Storage temperature	Tstg		-65 to 150	°C
Soldering temperature/time	Tsol		260°C, 10 seconds (lead section)	–

*1 In case of plastic package

26.2 Recommended Operating Conditions

(Ta = -20 to 70°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Operating power voltage	V_{DD}	Normal operation mode	1.8		3.6	V
		Flash programming mode	2.7		3.6	V
Operating frequency	f_{OSC3}	Crystal/ceramic oscillation	0.2		8.2	MHz
		CR oscillation	0.2		2.2	MHz
	f_{OSC1}	Crystal oscillation		32.768	100	kHz
Capacitor between V_{SS} and V_{D1} *1	C_1			0.1		µF
Capacitor between V_{SS} and V_{C1} *1	C_2			0.1		µF
Capacitor between V_{SS} and V_{C2} *1	C_3			0.1		µF
Capacitor between V_{SS} and V_{C3} *1	C_4			0.1		µF
Capacitor between V_{SS} and V_{C4} *1	C_5			0.1		µF
Capacitor between V_{SS} and V_{C5} *1	C_6			0.1		µF
Capacitor between CA and CB *1	C_7			0.1		µF
Capacitor between CA and CC *1	C_8			0.1		µF
Capacitor between CD and CE *1	C_9			0.1		µF
Capacitor between V_{SS} and V_{D2} *1	C_{10}			0.1		µF
Capacitor between CF and CG *1	C_{11}			0.1		µF

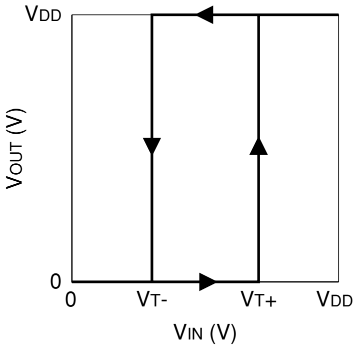
*1 The capacitors are not necessary when LCD driver is not used. In this case, leave the V_{C1} to V_{C5} and CA to CG pins open.

26.3 DC Characteristics

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -20 to 70°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level input voltage	V_{IH}	P*xx*	0.8V_{DD}		V_{DD}	V
Low level input voltage	V_{IL}	P*xx*	0		0.2V_{DD}	V
High level schmitt input voltage (1)	V_{T1+}	#RESET	0.5V_{DD}		0.9V_{DD}	V
Low level schmitt input voltage (1)	V_{T1-}	#RESET	0.1V_{DD}		0.5V_{DD}	V
High level schmitt input voltage (2) ∗1	V_{T2+}	P*xx*	0.5V_{DD}		0.9V_{DD}	V
Low level schmitt input voltage (2) ∗1	V_{T2-}	P*xx*	0.1V_{DD}		0.5V_{DD}	V
High level output current	I_{OH}	P*xx*, V_{OH} = 0.9V_{DD}			-0.5	mA
Low level output current	I_{OL}	P*xx*, V_{OL} = 0.1V_{DD}	0.5			mA
Input leak current	I_{LI}	P*xx*, #RESET	-1		1	µA
Output leak current	I_{LO}	P*xx*	-1		1	µA
Input pull-up resistance	R_{IN}	P*xx*, #RESET	100		500	kΩ
Input pin capacitance	C_{IN}	P*xx*, V_{IN} = 0V, f = 1MHz, Ta = 25°C			15	pF
Segment/Common output current	I_{SEGH}	SEG*xx*, COM*xx*, V_{SEGH} = V_{C5} - 0.1V			-5	µA
	I_{SEGL}	SEG*xx*, COM*xx*, V_{SEGL} = 0.1V	5			µA

∗1 When Schmitt input is enabled



26.4 Analog Circuit Characteristics

LCD driver

The typical values in the following LCD driver characteristics varies depending on the panel load (panel size, drive duty, number of display pixels and display contents), so evaluate them by connecting to the actually used LCD panel. See Section 26.8, “Characteristic Plots,” for the load characteristic.

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_1–C_{11} = 0.1μF, Checker pattern displayed,
No panel load

Item	Symbol	Condition		Min.	Typ.	Max.	Unit
LCD drive voltage	V_{C1}	Connect 1MΩ load resistor between V_{SS} and V_{C1}		0.18V_{C5}		0.22V_{C5}	V
	V_{C2}	Connect 1MΩ load resistor between V_{SS} and V_{C2}		0.39V_{C5}		0.43V_{C5}	V
	V_{C3}	Connect 1MΩ load resistor between V_{SS} and V_{C3}		0.59V_{C5}		0.63V_{C5}	V
	V_{C4}	Connect 1MΩ load resistor between V_{SS} and V_{C4}		0.79V_{C5}		0.83V_{C5}	V
	V_{C5}	Connect 1MΩ load resistor between V_{SS} and V_{C5}	LC[3:0] = 0x0	Typ. × 0.94	4.20	Typ. × 1.06	V
			LC[3:0] = 0x1		4.30		V
			LC[3:0] = 0x2		4.40		V
			LC[3:0] = 0x3		4.50		V
			LC[3:0] = 0x4		4.60		V
			LC[3:0] = 0x5		4.70		V
			LC[3:0] = 0x6		4.80		V
			LC[3:0] = 0x7		4.90		V
			LC[3:0] = 0x8		5.00		V
			LC[3:0] = 0x9		5.10		V
			LC[3:0] = 0xa		5.20		V
			LC[3:0] = 0xb		5.30		V
			LC[3:0] = 0xc		5.40		V
			LC[3:0] = 0xd		5.50		V
			LC[3:0] = 0xe		5.60		V
			LC[3:0] = 0xf		5.70		V

SVD

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
SVD voltage	V_{SVD}	SVDC[3:0] = 0x0		–		V
		SVDC[3:0] = 0x1		–		V
		SVDC[3:0] = 0x2		–		V
		SVDC[3:0] = 0x3	Typ. × 0.91	1.8	Typ. × 1.09	V
		SVDC[3:0] = 0x4		1.85		V
		SVDC[3:0] = 0x5		1.9		V
		SVDC[3:0] = 0x6		1.95		V
		SVDC[3:0] = 0x7		2.0		V
		SVDC[3:0] = 0x8		2.05		V
		SVDC[3:0] = 0x9		2.1		V
		SVDC[3:0] = 0xa		2.2		V
		SVDC[3:0] = 0xb		2.3		V
		SVDC[3:0] = 0xc		2.4		V
		SVDC[3:0] = 0xd		2.5		V
		SVDC[3:0] = 0xe		2.6		V
		SVDC[3:0] = 0xf		2.7		V
SVD response time	t_{SVD}				500	μs

Flash memory

Unless otherwise specified: V_{DD} = 2.7 to 3.6V (VD1MD = 1), V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Erase time *1	t_{SE}	Erase 4K bytes			25	ms
Programming time *1	t_{BP}	Program 16 bits			20	μs
Erase/program count *2	C_{FEP}		1000			times

*1 Data transfer and data verification are included and erase/program start control time is not included.

*2 The erase/program count assumes that “erasing + programming” or “programming only” is one count and the programmed data is guaranteed to be retained for 10 years.

26.5 Current Consumption

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_1–C_{11} = 0.1µF, No panel load, PCKEN = 3

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Current consumption in SLEEP mode	I_{SLP}	OSC1 = OFF, OSC3 = OFF, VD1MD = 0		1	2.5	µA
Current consumption in HALT mode	I_{HALT1}	OSC1 = 32kHz, OSC3 = OFF, VD1MD = 0, PCKEN = 0		2.6	5	µA
	I_{HALT2}	OSC1 = 32kHz, OSC3 = 8MHz (ceramic), VD1MD = 0		610	1200	µA
	I_{HALT3}	OSC1 = 32kHz, OSC3 = 2MHz (CR), VD1MD = 0		390	780	µA
Current consumption during execution *8	I_{EXE1}	OSC1 = 32kHz, OSC3 = OFF, VD1MD = 0, FLCYC = 4 (1.5 cycles)		14	20	µA
	I_{EXE2}	OSC1 = 32kHz, OSC3 = 8MHz (ceramic), VD1MD = 0, FLCYC = 1 (3.5 cycles)		1800	2700	µA
	I_{EXE2L}	OSC1 = 32kHz, OSC3 = 2MHz (ceramic), VD1MD = 0, FLCYC = 4 (1.5 cycles)		780	1170	µA
	I_{EXE3}	OSC1 = 32kHz, OSC3 = 2MHz (CR), VD1MD = 0		1000	2000	µA
	I_{EXE11}	OSC1 = 32kHz, OSC3 = OFF, VD1MD = 1, FLCYC = 4 (1.5 cycles)		32	50	µA
	I_{EXE21}	OSC1 = 32kHz, OSC3 = 8MHz (ceramic), VD1MD = 1, FLCYC = 1 (3.5 cycles)		3200	4800	µA
	I_{EXE31}	OSC1 = 32kHz, OSC3 = 2MHz (CR), VD1MD = 1		2100	4200	µA
Current consumption during execution in heavy load protection mode *8	I_{EXE1H}	OSC1 = 32kHz, OSC3 = OFF, VD1MD = 0, HVLD = 1, FLCYC = 4 (1.5 cycles)		19	30	µA
LCD circuit current *1	I_{LCD1}	DSPC[1:0] = 3 (all off), LC[3:0] = 0xf, OSC1 = 32kHz, V_{DD} = 2.5 to 3.6V		4	10	µA
LCD circuit current in heavy load protection mode *2	I_{LCD1H}	DSPC[1:0] = 3 (all off), LC[3:0] = 0xf, OSC1 = 32kHz, V_{DD} = 2.5 to 3.6V, LHVLD = 1		17	30	µA
LCD circuit current when the power voltage booster is active *3	I_{LCD2}	DSPC[1:0] = 3 (all off), LC[3:0] = 0xf, OSC1 = 32kHz, PBON = 1, V_{DD} = 1.8 to 2.5V		7	20	µA
LCD circuit current in heavy load protection mode when the power voltage booster is active *4	I_{LCD2H}	DSPC[1:0] = 3 (all off), LC[3:0] = 0xf, OSC1 = 32kHz, PBON = 1, V_{DD} = 1.8 to 2.5V, LHVLD = 1		35	60	µA
SVD circuit current *5	I_{SVD}	V_{DD} = 3.6V		5	10	µA
Flash memory erasing current *6	I_{FERS}	When the CPU runs with 8MHz clock, VD1MD = 1		7	14	mA
Flash memory programming current *7	I_{FPRG}	When the CPU runs with 8MHz clock, VD1MD = 1		7	14	mA

*1 This value is added to the current consumption in HALT mode or current consumption during execution when the LCD circuit is active. Current consumption increases according to the display contents and panel load.

*2 This value is added to the current consumption during execution in heavy load protection mode when the LCD circuit is active. Current consumption increases according to the display contents and panel load.

*3 This value is added to the current consumption in HALT mode or current consumption during execution when the power voltage booster and the LCD circuit are active. Current consumption increases according to the display contents and panel load.

*4 This value is added to the current consumption during execution in heavy load protection mode when the power voltage booster and the LCD circuit are active. Current consumption increases according to the display contents and panel load.

*5 This value is added to the current consumption during execution or current consumption during execution in heavy load protection mode when the SVD circuit is active.

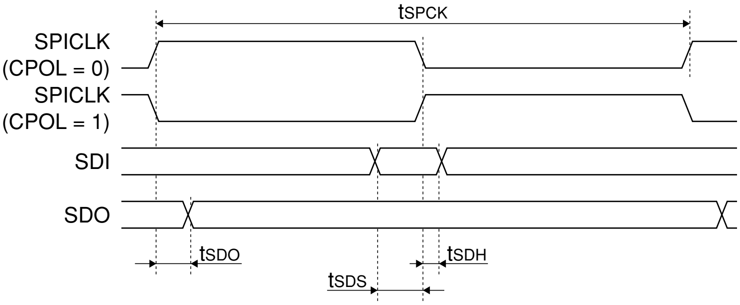
*6 This value is added to the current consumption during execution when the Flash memory is being erased in self-programming mode.

*7 This value is added to the current consumption during execution when the Flash memory is being programmed in self-programming mode.

*8 The values of current consumption while the CPU was operating were measured when a test program consisting of 60.5% ALU instructions, 17% branch instructions, 12% memory read instructions, and 10.5% memory write instructions was executed continuously in the Flash memory.

26.6 AC Characteristics

26.6.1 SPI AC Characteristics



Master mode

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -20 to 70°C

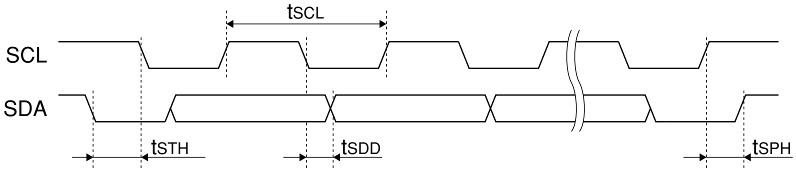
Item	Symbol	Min.	Typ.	Max.	Unit
SPICLK cycle time	t_{SPCK}	500			ns
SDI setup time	t_{SDS}	70			ns
SDI hold time	t_{SDH}	10			ns
SDO output delay time	t_{SDO}			20	ns

Slave mode

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -20 to 70°C

Item	Symbol	Min.	Typ.	Max.	Unit
SPICLK cycle time	t_{SPCK}	500			ns
SDI setup time	t_{SDS}	10			ns
SDI hold time	t_{SDH}	10			ns
SDO output delay time	t_{SDO}			80	ns

26.6.2 I²C AC Characteristics

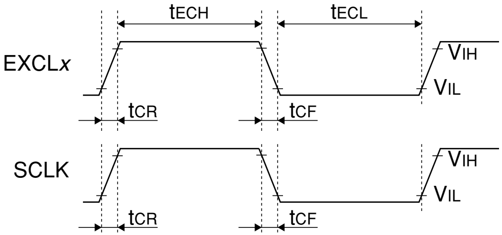


Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -20 to 70°C

Item	Symbol	Min.	Typ.	Max.	Unit
SCL cycle time	t_{SCL}	2500			ns
Start condition hold time	t_{STH}	$1/f_{SYS}$			ns
Data output delay time	t_{SDD}	$1/f_{SYS}$			ns
Stop condition hold time	t_{SPH}	$1/f_{SYS}$			ns

∗ f_{SYS}: System operating clock frequency

26.6.3 External Clock Input AC Characteristics

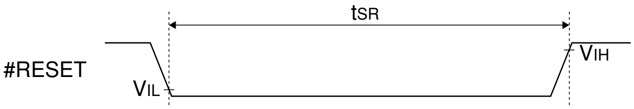


Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, V_{IH} = 0.8V_{DD}, V_{IL} = 0.2V_{DD}, Ta = -20 to 70°C

Item	Symbol	Min.	Typ.	Max.	Unit
EXCL*x* input High pulse width	t_{ECH}	2/f_{SYS}			s
EXCL*x* input Low pulse width	t_{ECL}	2/f_{SYS}			s
UART transfer rate	R_U			115200	bps
Input rise time	t_{CR}			80	ns
Input fall time	t_{CF}			80	ns

∗ f_{SYS}: System operating clock frequency

26.6.4 System AC Characteristics



Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, V_{IH} = 0.8V_{DD}, V_{IL} = 0.2V_{DD}, Ta = -20 to 70°C

Item	Symbol	Min.	Typ.	Max.	Unit
Reset Low pulse width	t_{SR}	100			μs

26.7 Oscillation Characteristics

Oscillation characteristics change depending on conditions (board pattern, components used, etc.). Use the following characteristics as reference values. In particular, when a ceramic resonator or crystal resonator is used for OSC3, use the resonator manufacturer's recommended values for constants such as capacitance and resistance. The oscillation start time is important because it becomes the wait time when OSC3 clock is used.

OSC1 (Crystal)

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, crystal resonator = C-002RX (R_1 = 30kΩ Typ.,
C_L = 12.5pF)*1, C_{G1} = 25pF external, C_{D1} = Built-in

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	t_{sta}				3	s
External gate capacitance	C_{G1}	Including board capacitance	0		25	pF
Built-in drain capacitance	C_{D1}	In case of the chip		10		pF
Frequency/IC deviation	∂f/∂IC	V_{DD} = constant	-10		10	ppm
Frequency/power voltage deviation	∂f/∂V				1	ppm/V
Frequency adjustment range	∂f/∂C_G	V_{DD} = constant, C_G = 0 to 25pF	25			ppm

*1 C-002RX: manufactured by Seiko Epson

OSC3 (Crystal)

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, crystal resonator = CA-301*1, R_f = 1MΩ,
C_{G3} = C_{D3} = 15pF

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time *2	t_{sta}				10	ms

*1 CA-301: manufactured by Seiko Epson

*2 The crystal oscillation start time changes by the crystal resonator to be used, C_{G3} and C_{D3}.

OSC3 (Ceramic)

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, ceramic resonator = KBR-4.0MSB/KBR-8.0MSB*1,
R_f = 1MΩ, C_{G3} = C_{D3} = 30pF

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time *2	t_{sta}				1	ms

*1 KBR-4.0MSB/KBR-8.0MSB: manufactured by Kyocera

*2 The ceramic oscillation start time changes by the ceramic resonator to be used, C_{G3} and C_{D3}.

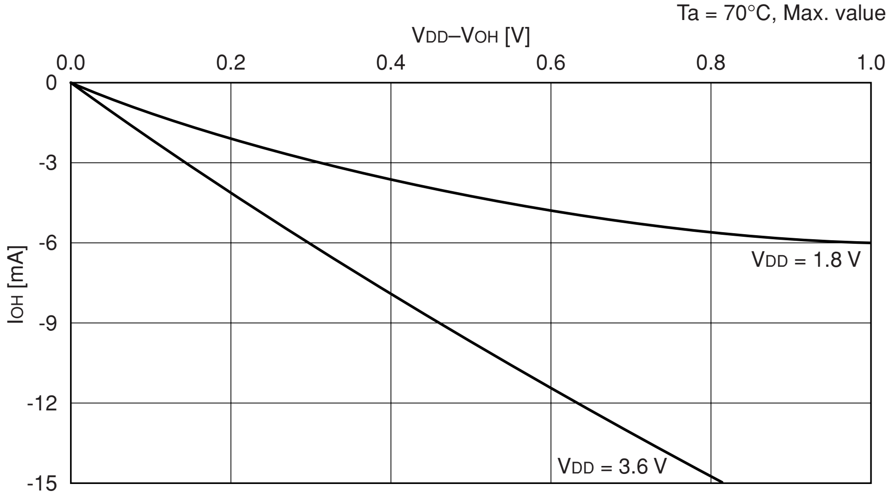
OSC3 (CR)

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C

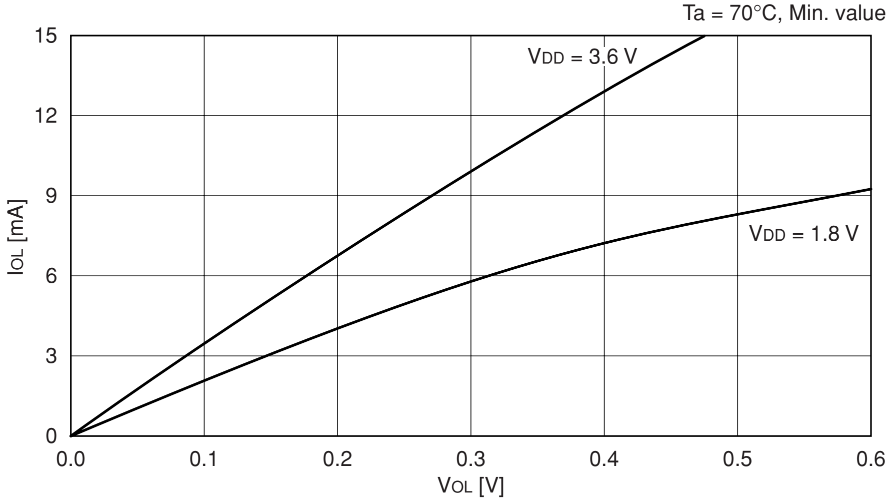
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	t_{sta}				100	µs
Frequency/IC deviation	∂f/∂IC	R_{CR} = constant	-25		25	%

26.8 Characteristic Plots (reference values)

High level output current - voltage characteristic

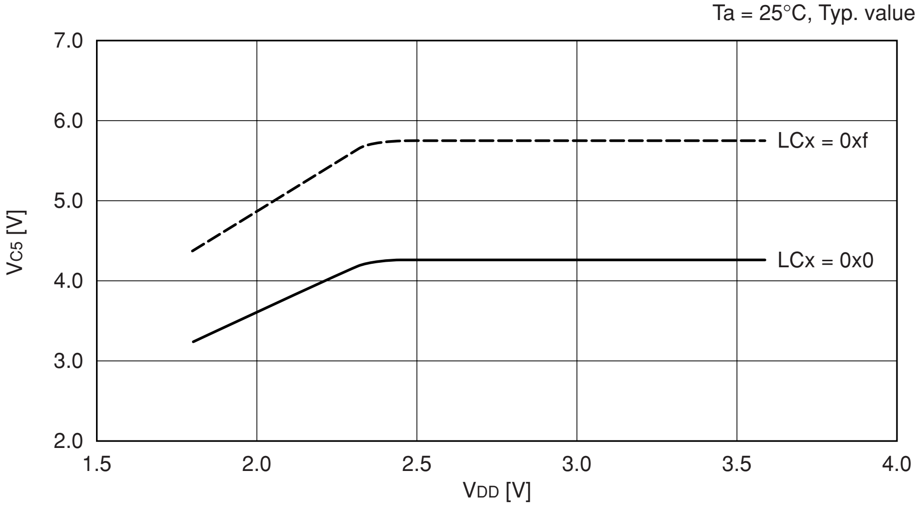


Low level output current - voltage characteristic



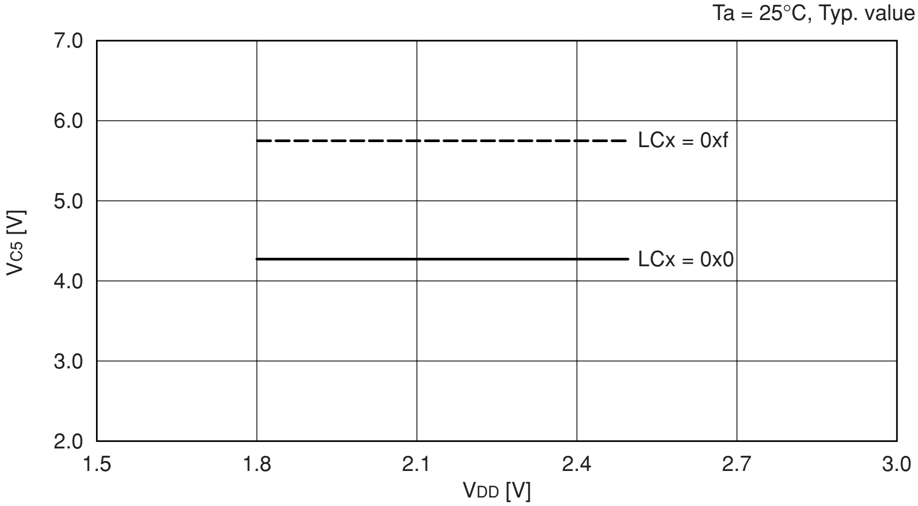
LCD drive voltage - supply voltage characteristic (without the power voltage booster used)

When a 1 MΩ load resistor is connected between V_{SS} and V_{C5}. (no panel load)

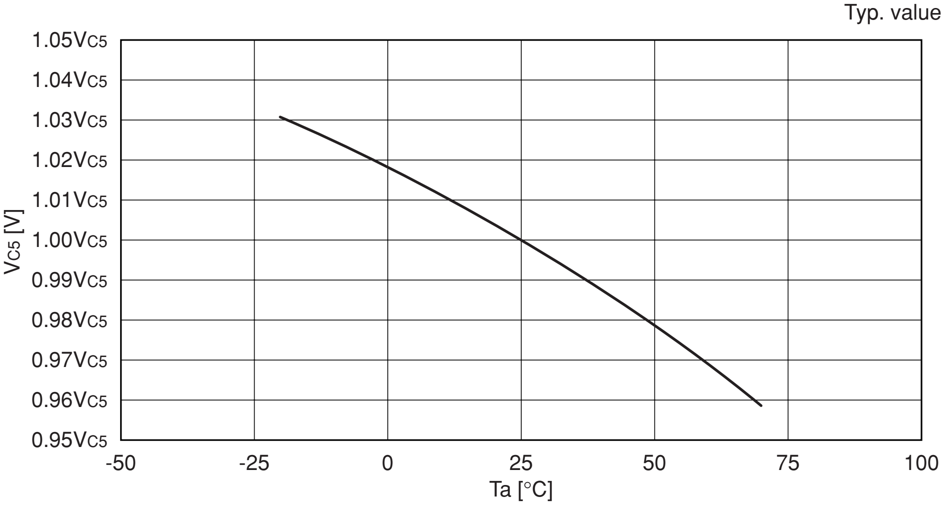


LCD drive voltage - supply voltage characteristic (with the power voltage booster used)

When a 1 MΩ load resistor is connected between V_{SS} and V_{C5}. (no panel load)

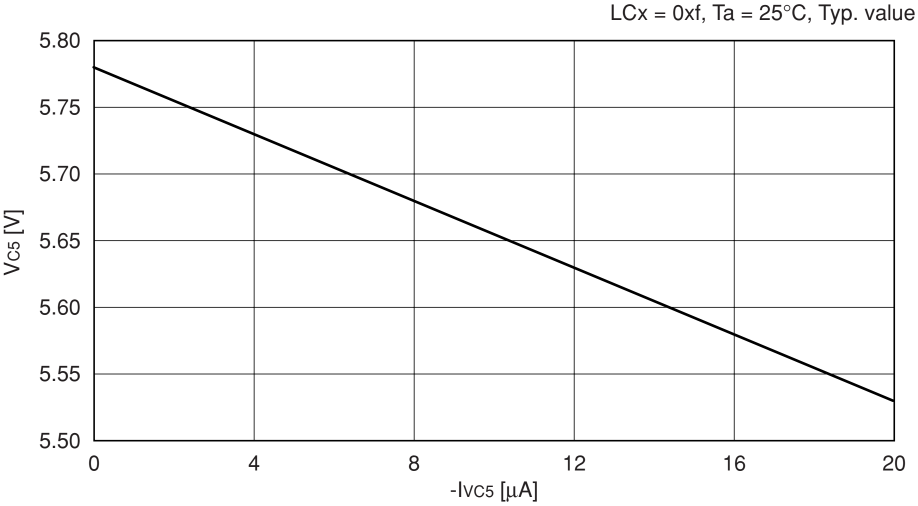


LCD drive voltage - ambient temperature characteristic



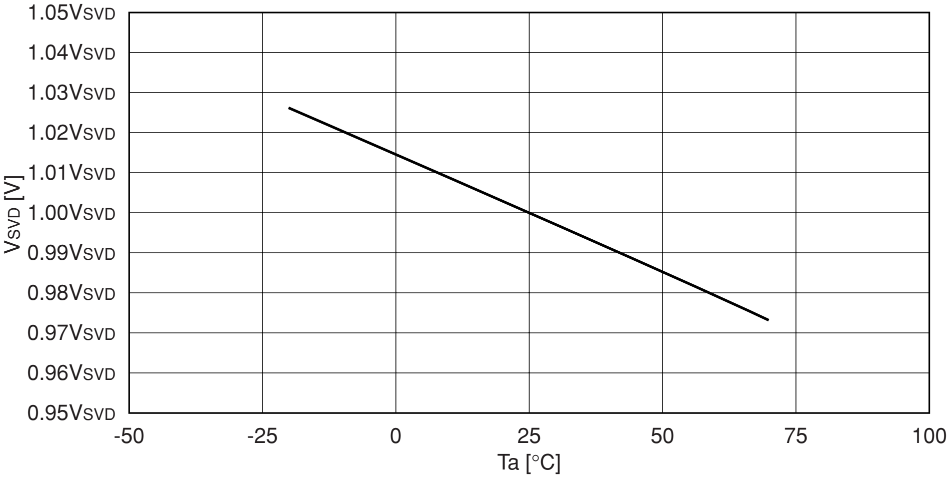
LCD drive voltage - load characteristic

When a load is connected to the V_{C5} pin only



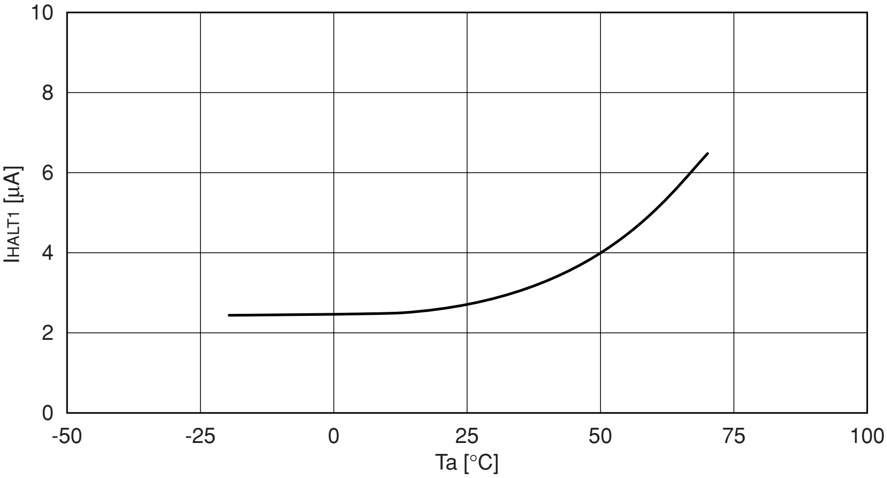
SVD voltage - ambient temperature characteristic

SVDCx = 0xf, Typ. value



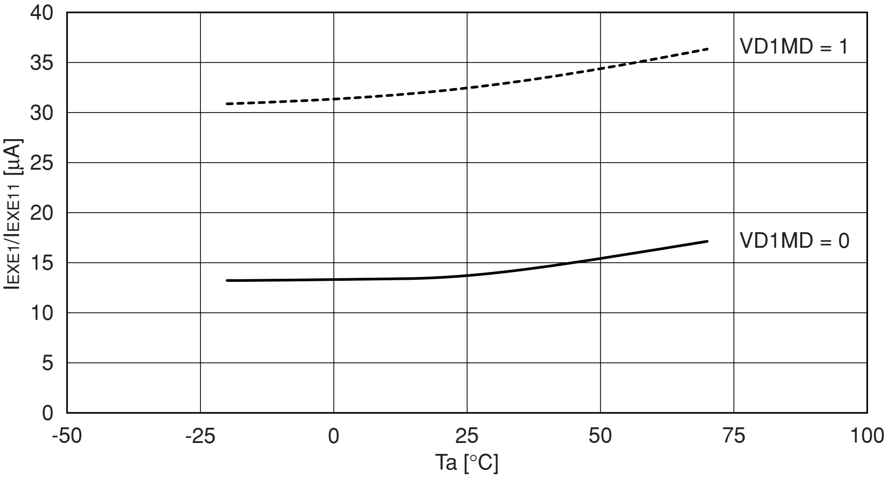
HALT state current consumption - temperature characteristic (during operation with OSC1)

<Crystal oscillation, fOSC1 = 32.768 kHz> OSC3 = OFF, VD1MD = 0, PCKEN = 0, Typ. value



Run state current consumption - temperature characteristic (during operation with OSC1)

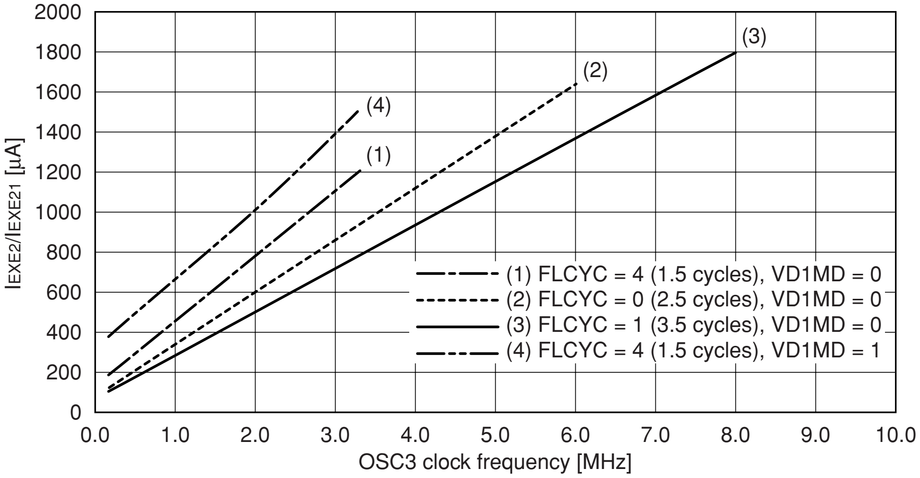
<Crystal oscillation, fOSC1 = 32.768 kHz> Typ. value



Run state current consumption - frequency characteristic (during operation with OSC3)

<Crystal oscillation/Ceramic oscillation>

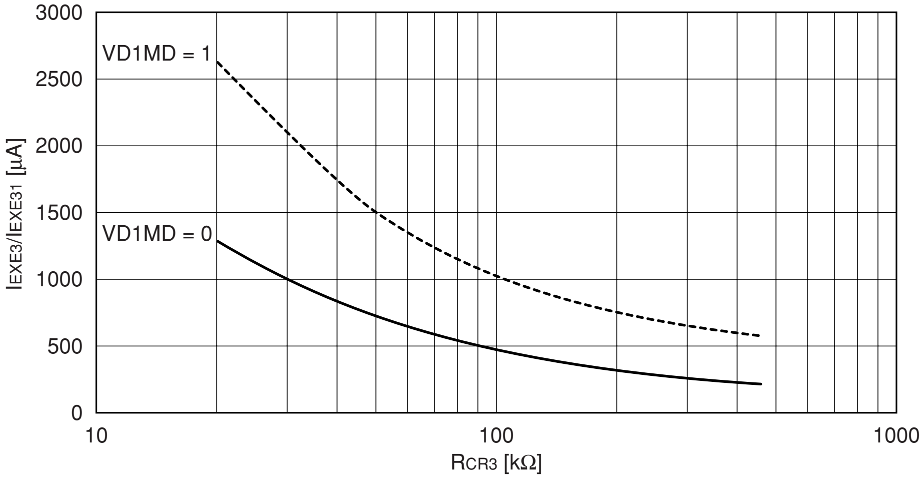
V_{DD} = 3.6 V, Ta = 25°C, Typ. value



Run status current consumption - resistor characteristic (during operation with OSC3)

<CR oscillation>

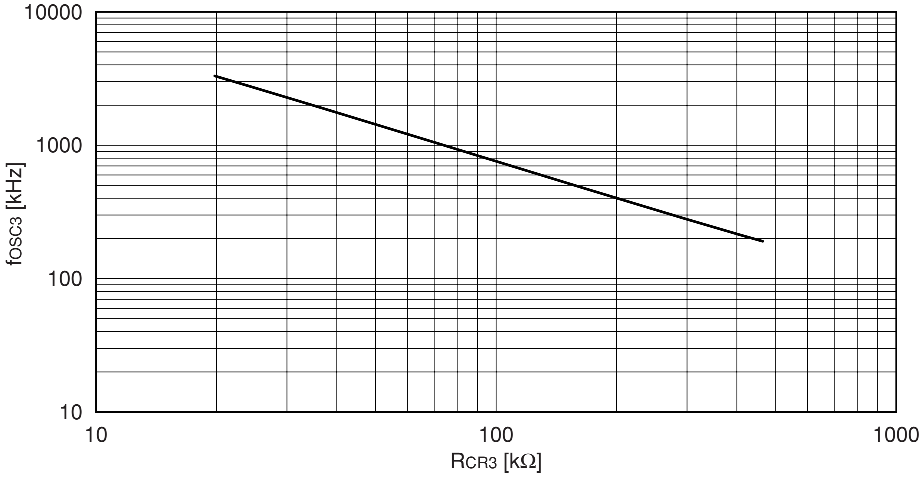
Ta = 25°C, Typ. value



OSC3 oscillation frequency - resistor characteristic

<CR oscillation>

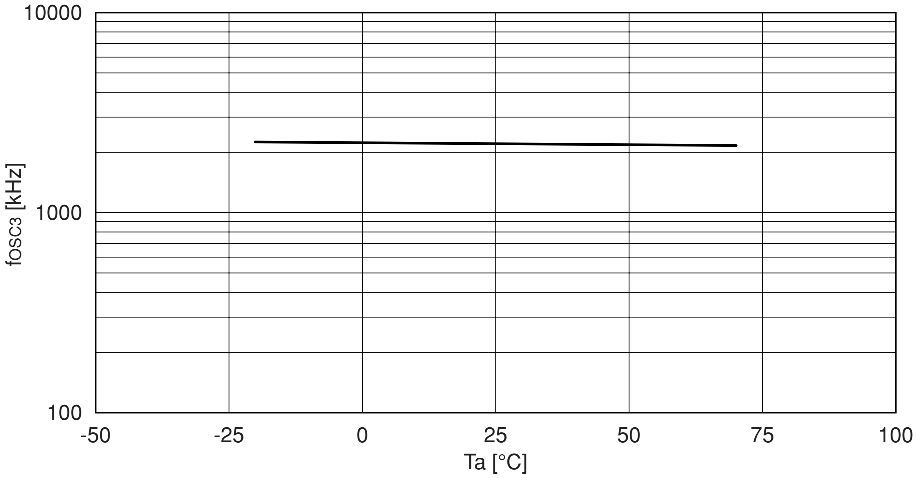
Ta = 25°C, Typ. value



OSC3 oscillation frequency - temperature characteristic

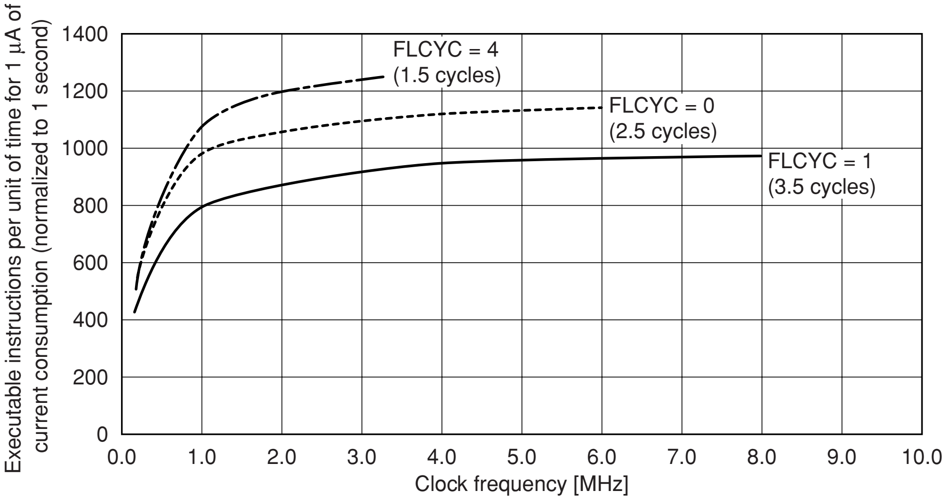
<CR oscillation>

R_{CR3} = 30 kΩ, Typ. value



Processing power - frequency characteristic

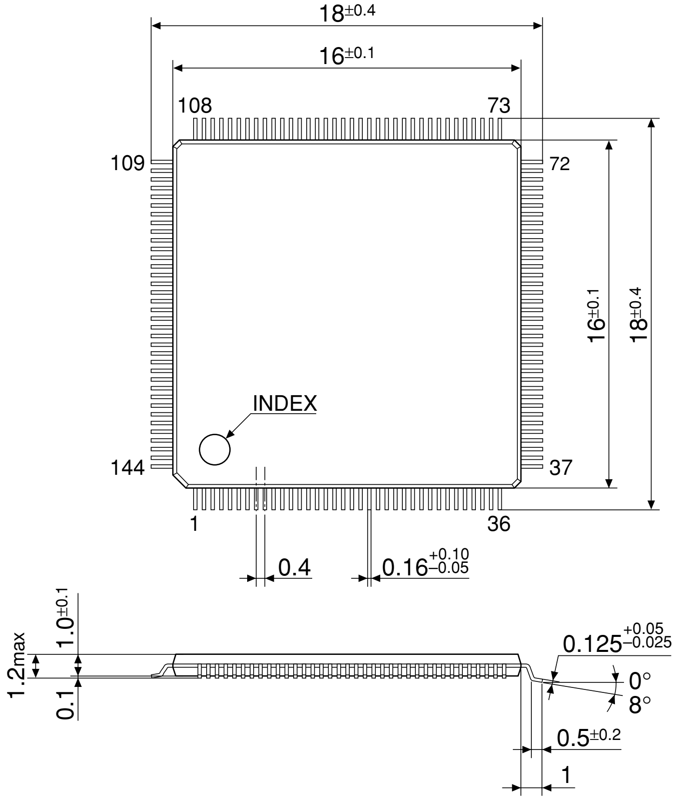
V_{DD} = 3.6V, VD1MD = 0, Typ. value



27 Package

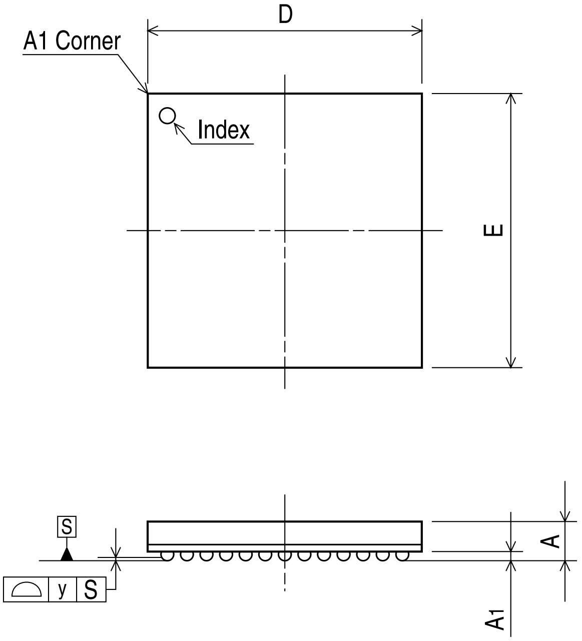
TQFP24-144-pin package

(Unit: mm)

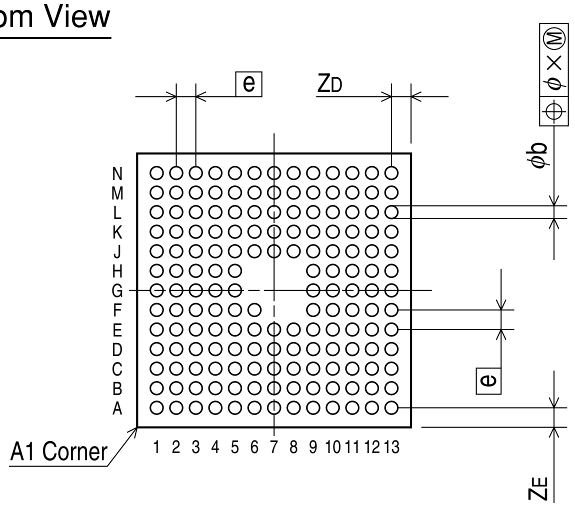


VFBGA7H-161 Package

Top View



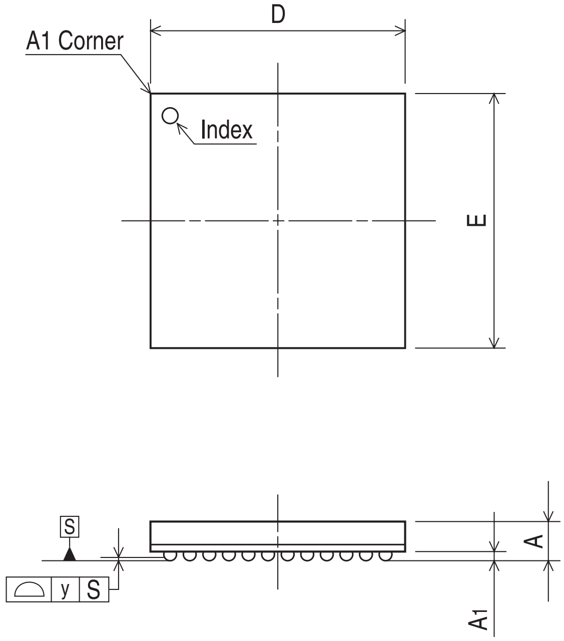
Bottom View



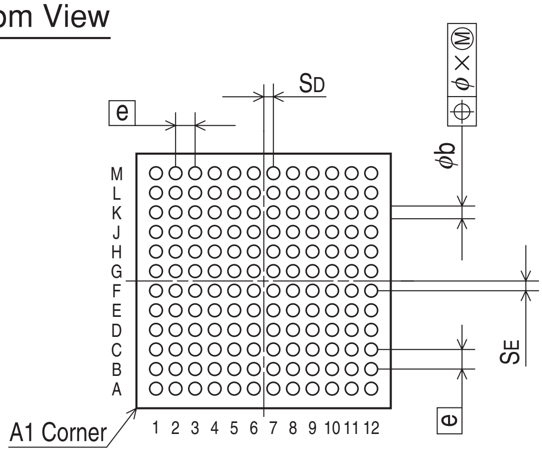
Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	–	7	–
E	–	7	–
A	–	–	1.0
A_1	–	0.23	–
$\boxed{e}$	–	0.5	–
b	0.26	–	0.36
X	–	–	0.08
y	–	–	0.1
Z_D	–	0.5	–
Z_E	–	0.5	–

VFBGA10H-144 Package

Top View



Bottom View

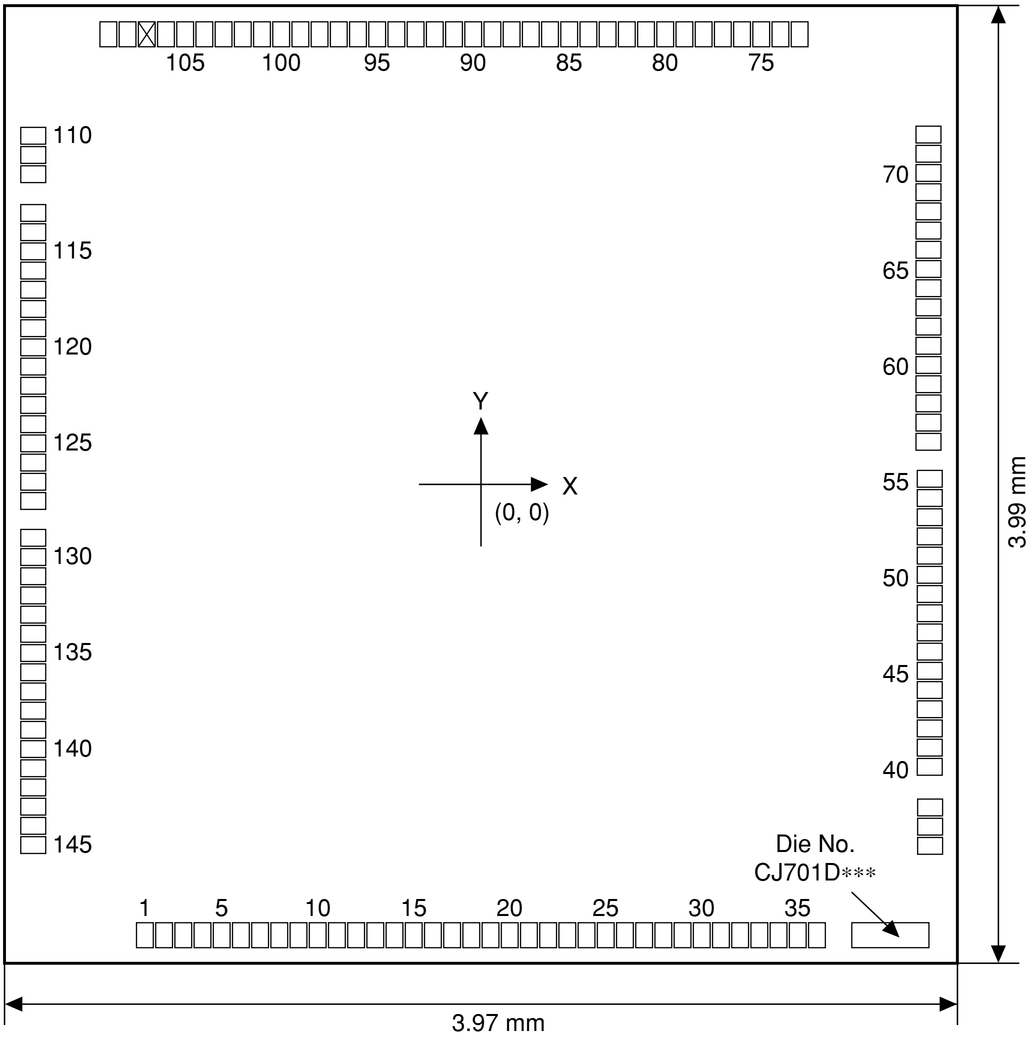


Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	–	10	–
E	–	10	–
A	–	–	1.0
A_1	–	0.3	–
$\boxed{e}$	–	0.8	–
b	0.38	–	0.48
X	–	–	0.08
y	–	–	0.1
S_D	–	0.4	–
S_E	–	0.4	–

THIS PAGE IS BLANK.

28 Pad Layout

28.1 Diagram of Pad Layout



Pad opening)	Pad No. 1–36, 73–109 :	70 × 104 µm
	Pad No. 37–72, 110–145:	104 × 70 µm
Chip thickness)	400 µm	

28.2 Pad Coordinates

(Unit: mm)

No.	Name	X	Y	No.	Name	X	Y	No.	Name	X	Y	No.	Name	X	Y
1	SEG17	-1.400	-1.880	37	SEG53	1.870	-1.505	73	V_{DD}	1.325	1.880	110	P02	-1.870	1.455
2	SEG18	-1.320	-1.880	38	SEG54	1.870	-1.425	74	OSC3	1.245	1.880	111	P01	-1.870	1.375
3	SEG19	-1.240	-1.880	39	SEG55	1.870	-1.345	75	OSC4	1.165	1.880	112	P00	-1.870	1.295
4	SEG20	-1.160	-1.880	40	COM31/SEG56	1.870	-1.185	76	V_{SS}	1.085	1.880	113	COM0	-1.870	1.135
5	SEG21	-1.080	-1.880	41	COM30/SEG57	1.870	-1.105	77	V_{D1}	1.005	1.880	114	COM1	-1.870	1.055
6	SEG22	-1.000	-1.880	42	COM29/SEG58	1.870	-1.025	78	OSC1	0.925	1.880	115	COM2	-1.870	0.975
7	SEG23	-0.920	-1.880	43	COM28/SEG59	1.870	-0.945	79	OSC2	0.845	1.880	116	COM3	-1.870	0.895
8	SEG24	-0.840	-1.880	44	COM27/SEG60	1.870	-0.865	80	#TEST	0.765	1.880	117	COM4	-1.870	0.815
9	SEG25	-0.760	-1.880	45	COM26/SEG61	1.870	-0.785	81	#RESET	0.685	1.880	118	COM5	-1.870	0.735
10	SEG26	-0.680	-1.880	46	COM25/SEG62	1.870	-0.705	82	DSIO/P33	0.605	1.880	119	COM6	-1.870	0.655
11	SEG27	-0.600	-1.880	47	COM24/SEG63	1.870	-0.625	83	DST2/P32	0.525	1.880	120	COM7	-1.870	0.575
12	SEG28	-0.520	-1.880	48	COM23/SEG64	1.870	-0.545	84	DCLK/P31	0.445	1.880	121	COM8	-1.870	0.495
13	SEG29	-0.440	-1.880	49	COM22/SEG65	1.870	-0.465	85	P30/FOUT3	0.365	1.880	122	COM9	-1.870	0.415
14	SEG30	-0.360	-1.880	50	COM21/SEG66	1.870	-0.385	86	P27/EXCL3	0.285	1.880	123	COM10	-1.870	0.335
15	SEG31	-0.280	-1.880	51	COM20/SEG67	1.870	-0.305	87	P26/TOUT	0.205	1.880	124	COM11	-1.870	0.255
16	SEG32	-0.200	-1.880	52	COM19/SEG68	1.870	-0.225	88	P25/SCLK	0.125	1.880	125	COM12	-1.870	0.175
17	SEG33	-0.120	-1.880	53	COM18/SEG69	1.870	-0.145	89	P24/SOUT	0.045	1.880	126	COM13	-1.870	0.095
18	SEG34	-0.040	-1.880	54	COM17/SEG70	1.870	-0.065	90	P23/SIN	-0.035	1.880	127	COM14	-1.870	0.015
19	SEG35	0.040	-1.880	55	COM16/SEG71	1.870	0.015	91	P22/SPICLK	-0.115	1.880	128	COM15	-1.870	-0.065
20	SEG36	0.120	-1.880	56	V_{SS}	1.870	0.175	92	P21/SDO	-0.195	1.880	129	SEG0	-1.870	-0.225
21	SEG37	0.200	-1.880	57	TEST1	1.870	0.255	93	P20/SDI	-0.275	1.880	130	SEG1	-1.870	-0.305
22	SEG38	0.280	-1.880	58	TEST2	1.870	0.335	94	P17/#SPISS	-0.355	1.880	131	SEG2	-1.870	-0.385
23	SEG39	0.360	-1.880	59	TEST3	1.870	0.415	95	P16/EXCL0	-0.435	1.880	132	SEG3	-1.870	-0.465
24	SEG40	0.440	-1.880	60	V_{D2}	1.870	0.495	96	P15/SCL	-0.515	1.880	133	SEG4	-1.870	-0.545
25	SEG41	0.520	-1.880	61	CG	1.870	0.575	97	P14/SDA	-0.595	1.880	134	SEG5	-1.870	-0.625
26	SEG42	0.600	-1.880	62	CF	1.870	0.655	98	P07/EXCL1	-0.675	1.880	135	SEG6	-1.870	-0.705
27	SEG43	0.680	-1.880	63	CE	1.870	0.735	99	P06/EXCL2	-0.755	1.880	136	SEG7	-1.870	-0.785
28	SEG44	0.760	-1.880	64	CD	1.870	0.815	100	P05/REMO	-0.835	1.880	137	SEG8	-1.870	-0.865
29	SEG45	0.840	-1.880	65	CC	1.870	0.895	101	P04/REMI	-0.915	1.880	138	SEG9	-1.870	-0.945
30	SEG46	0.920	-1.880	66	CB	1.870	0.975	102	P13/FOUT1	-0.995	1.880	139	SEG10	-1.870	-1.025
31	SEG47	1.000	-1.880	67	CA	1.870	1.055	103	P12	-1.075	1.880	140	SEG11	-1.870	-1.105
32	SEG48	1.080	-1.880	68	V_{C5}	1.870	1.135	104	P11	-1.155	1.880	141	SEG12	-1.870	-1.185
33	SEG49	1.160	-1.880	69	V_{C4}	1.870	1.215	105	P10	-1.235	1.880	142	SEG13	-1.870	-1.265
34	SEG50	1.240	-1.880	70	V_{C3}	1.870	1.295	106	P03	-1.315	1.880	143	SEG14	-1.870	-1.345
35	SEG51	1.320	-1.880	71	V_{C2}	1.870	1.375	107	N.C.	-1.395	1.880	144	SEG15	-1.870	-1.425
36	SEG52	1.400	-1.880	72	V_{C1}	1.870	1.455	108	V_{DD}	-1.475	1.880	145	SEG16	-1.870	-1.505
–	–	–	–	–	–	–	–	109	V_{SS}	-1.555	1.880	–	–	–	–

Appendix A List of I/O Registers

Periperal	Address	Register name		Function
Prescaler (8-bit device)	0x4020	PSC_CTL	Prescaler Control Register	Starts/stops the prescaler.
	0x4021–0x403f	–	–	Reserved
UART (with IrDA) (8-bit device)	0x4100	UART_ST	UART Status Register	Indicates transfer, buffer and error statuses.
	0x4101	UART_TXD	UART Transmit Data Register	Transmit data
	0x4102	UART_RXD	UART Receive Data Register	Receive data
	0x4103	UART_MOD	UART Mode Register	Sets transfer data format.
	0x4104	UART_CTL	UART Control Register	Controls data transfer.
	0x4105	UART_EXP	UART Expansion Register	Sets IrDA mode.
	0x4106–0x411f	–	–	Reserved
8-bit timer (with fine mode) (16-bit device)	0x4200	T8F_CLK	8-bit Timer Input Clock Select Register	Selects a prescaler output clock.
	0x4202	T8F_TR	8-bit Timer Reload Data Register	Sets reload data.
	0x4204	T8F_TC	8-bit Timer Counter Data Register	Counter data
	0x4206	T8F_CTL	8-bit Timer Control Register	Sets the timer mode and starts/stops the timer.
	0x4208–0x421f	–	–	Reserved
16-bit timer Ch. 0 (16-bit device)	0x4220	T16_CLK0	16-bit Timer Ch.0 Input Clock Select Register	Selects a prescaler output clock.
	0x4222	T16_TR0	16-bit Timer Ch.0 Reload Data Register	Sets reload data.
	0x4224	T16_TC0	16-bit Timer Ch.0 Counter Data Register	Counter data
	0x4226	T16_CTL0	16-bit Timer Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
	0x4228–0x423f	–	–	Reserved
16-bit timer Ch. 1 (16-bit device)	0x4240	T16_CLK1	16-bit Timer Ch.1 Input Clock Select Register	Selects a prescaler output clock.
	0x4242	T16_TR1	16-bit Timer Ch.1 Reload Data Register	Sets reload data.
	0x4244	T16_TC1	16-bit Timer Ch.1 Counter Data Register	Counter data
	0x4246	T16_CTL1	16-bit Timer Ch.1 Control Register	Sets the timer mode and starts/stops the timer.
	0x4248–0x425f	–	–	Reserved
16-bit timer Ch. 2 (16-bit device)	0x4260	T16_CLK2	16-bit Timer Ch.2 Input Clock Select Register	Selects a prescaler output clock.
	0x4262	T16_TR2	16-bit Timer Ch.2 Reload Data Register	Sets reload data.
	0x4264	T16_TC2	16-bit Timer Ch.2 Counter Data Register	Counter data
	0x4266	T16_CTL2	16-bit Timer Ch.2 Control Register	Sets the timer mode and starts/stops the timer.
	0x4268–0x427f	–	–	Reserved
Interrupt controller (16-bit device)	0x4300	ITC_IFLG	Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x4302	ITC_EN	Interrupt Enable Register	Enables/disables each maskable interrupt.
	0x4304	ITC_CTL	ITC Control Register	Enables/disables the ITC.
	0x4306	ITC_ELV0	External Interrupt Level Setup Register 0	Sets the P0 and P1 interrupt levels and trigger modes.
	0x4308	ITC_ELV1	External Interrupt Level Setup Register 1	Sets the stopwatch timer and clock timer interrupt levels and trigger modes.
	0x430a	ITC_ELV2	External Interrupt Level Setup Register 2	Sets the 8-bit OSC1 timer and SVD interrupt levels and trigger modes.
	0x430c	ITC_ELV3	External Interrupt Level Setup Register 3	Sets the LCD and PWM & capture timer interrupt levels and trigger modes.
	0x430e	ITC_ILV0	Internal Interrupt Level Setup Register 0	Sets the 8-bit timer and 16-bit timer Ch. 0 interrupt levels.
	0x4310	ITC_ILV1	Internal Interrupt Level Setup Register 1	Sets the 16-bit timer Ch. 1 and 16-bit timer Ch. 2 interrupt levels.
	0x4312	ITC_ILV2	Internal Interrupt Level Setup Register 2	Sets the UART and remote controller interrupt levels.
	0x4314	ITC_ILV3	Internal Interrupt Level Setup Register 3	Sets the SPI and I^{2}C interrupt levels.
	0x4316–0x431f	–	–	Reserved
SPI (16-bit device)	0x4320	SPI_ST	SPI Status Register	Indicates transfer and buffer statuses.
	0x4322	SPI_TXD	SPI Transmit Data Register	Transmit data
	0x4324	SPI_RXD	SPI Receive Data Register	Receive data
	0x4326	SPI_CTL	SPI Control Register	Sets the SPI mode and enables data transfer.
	0x4328–0x433f	–	–	Reserved
I^{2}C (16-bit device)	0x4340	I2C_EN	I^{2}C Enable Register	Enables the I^{2}C module.
	0x4342	I2C_CTL	I^{2}C Control Register	Controls the I^{2}C operation and indicates transfer status.
	0x4344	I2C_DAT	I^{2}C Data Register	Transmit/receive data
	0x4346	I2C_ICTL	I^{2}C Interrupt Control Register	Controls the I^{2}C interrupt.
	0x4348–0x435f	–	–	Reserved

Periperal	Address	Register name		Function
Clock timer (8-bit device)	0x5000	CT_CTL	Clock Timer Control Register	Resets and starts/stops the timer.
	0x5001	CT_CNT	Clock Timer Counter Register	Counter data
	0x5002	CT_IMSK	Clock Timer Interrupt Mask Register	Enables/disables interrupt.
	0x5003	CT_IFLG	Clock Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x5004–0x501f	–	–	Reserved
Stopwatch timer (8-bit device)	0x5020	SWT_CTL	Stopwatch Timer Control Register	Resets and starts/stops the timer.
	0x5021	SWT_BCNT	Stopwatch Timer BCD Counter Register	BCD counter data
	0x5022	SWT_IMSK	Stopwatch Timer Interrupt Mask Register	Enables/disables interrupt.
	0x5023	SWT_IFLG	Stopwatch Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x5024–0x503f	–	–	Reserved
Watchdog timer (8-bit device)	0x5040	WDT_CTL	Watchdog Timer Control Register	Resets and starts/stops the timer.
	0x5041	WDT_ST	Watchdog Timer Status Register	Sets the timer mode and indicates NMI status.
	0x5042–0x505f	–	–	Reserved
Oscillator (8-bit device)	0x5060	OSC_SRC	Clock Source Select Register	Selects a clock source.
	0x5061	OSC_CTL	Oscillation Control Register	Controls oscillation.
	0x5062	OSC_NFEN	Noise Filter Enable Register	Enables/disables noise filters.
	0x5063	OSC_LCLK	LCD Clock Setup Register	Sets up the LCD clock
	0x5064	OSC_FOUT	FOUT Control Register	Controls clock output.
	0x5065	OSC_T8OSC1	T8OSC1 Clock Control Register	Sets up the 8-bit OSC1 timer clock.
	0x5066–0x507f	–	–	Reserved
Clock generator (8-bit device)	0x5080	CLG_PCLK	PCLK Control Register	Controls the PCLK output.
	0x5081	CLG_CCLK	CCLK Control Register	Configures the CCLK division ratio
	0x5082–0x509f	–	–	Reserved
LCD driver (8-bit device)	0x50a0	LCD_DCTL	LCD Display Control Register	Controls the LCD display
	0x50a1	LCD_CADJ	LCD Contrast Adjust Register	Controls the contrast.
	0x50a2	LCD_CCTL	LCD Clock Control Register	Controls the LCD clock duty.
	0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	Controls the LCD drive voltage regulator.
	0x50a4	LCD_PWR	LCD Power Voltage Booster Control Register	Controls the LCD voltage booster.
	0x50a5	LCD_IMSK	LCD Interrupt Mask Register	Enables/disables interrupt.
	0x50a6	LCD_IFLG	LCD Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x50a7–0x50bf	–	–	Reserved
8-bit OSC1 timer (8-bit device)	0x50c0	T8OSC1_CTL	8-bit OSC1 Timer Control Register	Sets the timer mode and starts/stops the timer.
	0x50c1	T8OSC1_CNT	8-bit OSC1 Timer Counter Data Register	Counter data
	0x50c2	T8OSC1_CMP	8-bit OSC1 Timer Compare Data Register	Sets compare data.
	0x50c3	T8OSC1_IMSK	8-bit OSC1 Timer Interrupt Mask Register	Enables/disables interrupt.
	0x50c4	T8OSC1_IFLG	8-bit OSC1 Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x50c5–0x50df	–	–	Reserved
SVD circuit (8-bit device)	0x5100	SVD_EN	SVD Enable Register	Enables/disables the SVD operation.
	0x5101	SVD_CMP	SVD Compare Voltage Register	Sets compare voltage.
	0x5102	SVD_RSLT	SVD Detection Result Register	Voltage detection results
	0x5103	SVD_IMSK	SVD Interrupt Mask Register	Enables/disables interrupt.
	0x5104	SVD_IFLG	SVD Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x5105–0x511f	–	–	Reserved
Power supply circuit (8-bit device)	0x5120	VD1_CTL	V_{D1} Control Register	Controls the V_{D1} voltage and heavy load protection mode.
	0x5121–0x513f	–	–	Reserved
P port & port MUX (8-bit device)	0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
	0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
	0x5202	P0_IO	P0 Port I/O Direction Control Register	Selects the P0 port I/O direction.
	0x5203	P0_PU	P0 Port Pull-up Control Register	Controls the P0 port pull-up resistor.
	0x5204	P0_SM	P0 Port Schmitt Trigger Control Register	Controls the P0 port Schmitt trigger input.
	0x5205	P0_IMSK	P0 Port Interrupt Mask Register	Enables/disables the P0 port interrupt.
	0x5206	P0_EDGE	P0 Port Interrupt Edge Select Register	Selects the signal edge for generating P0 port interrupts
	0x5207	P0_IFLG	P0 Port Interrupt Flag Register	Indicates/resets the P0 port interrupt occurrence status.
	0x5208	P0_CHAT	P0 Port Chattering Filter Control Register	Controls the P0 port chattering filter.
	0x5209	P0_KRST	P0 Port Key-Entry Reset Configuration Register	Configures the P0 port key-entry reset function.
	0x520a–0x520f	–	–	Reserved
	0x5210	P1_IN	P1 Port Input Data Register	P1 port input data
	0x5211	P1_OUT	P1 Port Output Data Register	P1 port output data
	0x5212	P1_IO	P1 Port I/O Direction Control Register	Selects the P1 port I/O direction.
	0x5213	P1_PU	P1 Port Pull-up Control Register	Controls the P1 port pull-up resistor.
	0x5214	P1_SM	P1 Port Schmitt Trigger Control Register	Controls the P1 port Schmitt trigger input.
	0x5215	P1_IMSK	P1 Port Interrupt Mask Register	Enables/disables the P1 port interrupt.
	0x5216	P1_EDGE	P1 Port Interrupt Edge Select Register	Selects the signal edge for generating P1 port interrupts

Periperal	Address	Register name		Function
P port & port MUX (8-bit device)	0x5217	P1_IFLG	P1 Port Interrupt Flag Register	Indicates/resets the P1 port interrupt occurrence status.
	0x5218–0x521f	–	–	Reserved
	0x5220	P2_IN	P2 Port Input Data Register	P2 port input data
	0x5221	P2_OUT	P2 Port Output Data Register	P2 port output data
	0x5222	P2_IO	P2 Port I/O Direction Control Register	Selects the P2 port I/O direction.
	0x5223	P2_PU	P2 Port Pull-up Control Register	Controls the P2 port pull-up resistor.
	0x5224	P2_SM	P2 Port Schmitt Trigger Control Register	Controls the P2 port Schmitt trigger input.
	0x5225–0x522f	–	–	Reserved
	0x5230	P3_IN	P3 Port Input Data Register	P3 port input data
	0x5231	P3_OUT	P3 Port Output Data Register	P3 port output data
	0x5232	P3_IO	P3 Port I/O Direction Control Register	Selects the P3 port I/O direction.
	0x5233	P3_PU	P3 Port Pull-up Control Register	Controls the P3 port pull-up resistor.
	0x5234	P3_SM	P3 Port Schmitt Trigger Control Register	Controls the P3 port Schmitt trigger input.
	0x5235–0x527f	–	–	Reserved
	0x52a0	P0_PMUX	P0 Port Function Select Register	Selects the P0 port function.
	0x52a1	P1_PMUX	P1 Port Function Select Register	Selects the P1 port function.
	0x52a2	P2_PMUX	P2 Port Function Select Register	Selects the P2 port function.
	0x52a3	P3_PMUX	P3 Port Function Select Register	Selects the P3 port function.
	0x52a4–0x52bf	–	–	Reserved
PWM & capture timer (16-bit device)	0x5300	T16E_CA	PWM Timer Compare Data A Register	Sets compare data A.
	0x5302	T16E_CB	PWM Timer Compare Data B Register	Sets compare data B.
	0x5304	T16E_TC	PWM Timer Counter Data Register	Counter data
	0x5306	T16E_CTL	PWM Timer Control Register	Sets the timer mode and starts/stops the timer.
	0x5308	T16E_CLK	PWM Timer Input Clock Select Register	Selects a prescaler output clock.
	0x530a	T16E_IMSK	PWM Timer Interrupt Mask Register	Enables/disables interrupt.
	0x530c	T16E_IFLG	PWM Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x530e–0x531f	–	–	Reserved
MISC register (8-bit device)	0x5320	MISC_FL	FLASHC Control Register	Sets FLASHC access condition.
	0x5321	MISC_SR	SRAMC Control Register	Sets the SRAMC access condition.
	0x5322	MISC_OSC1	OSC1 Peripheral Control Register	Selects the OSC1 peripheral operation in debug mode.
	0x5323–0x533f	–	–	Reserved
Remote controller (8-bit device)	0x5340	REMC_CFG	REMC Configuration Register	Selects/enables transmission/reception
	0x5341	REMC_PSC	REMC Prescaler Clock Select Register	Selects a prescaler output clock.
	0x5342	REMC_CARH	REMC H Carrier Length Setup Register	Sets up the H period of the carrier.
	0x5343	REMC_CARL	REMC L Carrier Length Setup Register	Sets up the L period of the carrier.
	0x5344	REMC_ST	REMC Status Register	Transmit/receive bit
	0x5345	REMC_LCNT	REMC Length Counter Register	Sets the transmit/receive data length.
	0x5346	REMC_IMSK	REMC Interrupt Mask Register	Enables/disables interrupt.
	0x5347	REMC_IFLG	REMC Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x5348–0x535f	–	–	Reserved
S1C17 Core I/O	0xffff80	TTBR	Vector Table Base Register	Indicates the vector table base address.
	0xffff84	IDIR	Processor ID Register	Indicates the processor ID.
	0xffff90	DBRAM	Debug RAM Base Register	Indicates the debug RAM base address.

Note: Do not access the "Reserved" address in the table above and unused areas in the peripheral area that are not described in the table from the application program.

0x4020 Prescaler

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Prescaler Control Register (PSC_CTL)	**0x4020** (8 bits)	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**PRUND**	Prescaler run/stop in debug mode	1	Run	0	Stop	0	R/W	
		D0	**PRUN**	Prescaler run/stop control	1	Run	0	Stop	0	R/W	

0x4100–0x4105 UART (with IrDA)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Status Register (UART_ST)	**0x4100** (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		D6	**FER**	Framing error flag	1	Error	0	Normal	0	R/W	Reset by writing 1.
		D5	**PER**	Parity error flag	1	Error	0	Normal	0	R/W	
		D4	**OER**	Overrun error flag	1	Error	0	Normal	0	R/W	
		D3	**RD2B**	Second byte receive flag	1	Ready	0	Empty	0	R	
		D2	**TRBS**	Transmit busy flag	1	Busy	0	Idle	0	R	Shift register status
		D1	**RDRY**	Receive data ready flag	1	Ready	0	Empty	0	R	
		D0	**TDBE**	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	
UART Transmit Data Register (UART_TXD)	**0x4101** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)				0x0	R/W	
UART Receive Data Register (UART_RXD)	**0x4102** (8 bits)	D7–0	**RXD[7:0]**	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)				0x0	R	Older data in the buffer is read out first.
UART Mode Register (UART_MOD)	**0x4103** (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	**CHLN**	Character length	1	8 bits	0	7 bits	0	R/W	
		D3	**PREN**	Parity enable	1	With parity	0	No parity	0	R/W	
		D2	**PMD**	Parity mode select	1	Odd	0	Even	0	R/W	
		D1	**STPB**	Stop bit select	1	2 bits	0	1 bit	0	R/W	
		D0	**SSCK**	Input clock select	1	External	0	Internal	0	R/W	
UART Control Register (UART_CTL)	**0x4104** (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		D6	**REIEN**	Receive error int. enable	1	Enable	0	Disable	0	R/W	
		D5	**RIEN**	Receive buffer full int. enable	1	Enable	0	Disable	0	R/W	
		D4	**TIEN**	Transmit buffer empty int. enable	1	Enable	0	Disable	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	**RBFI**	Receive buffer full int. condition	1	2 bytes	0	1 byte	0	R/W	
		D0	**RXEN**	UART enable	1	Enable	0	Disable	0	R/W	
UART Expansion Register (UART_EXP)	**0x4105** (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock select	IRCLK[2:0]		Clock		0x0	R/W	
					0x7		PCLK•1/128				
					0x6		PCLK•1/64				
					0x5		PCLK•1/32				
					0x4		PCLK•1/16				
					0x3		PCLK•1/8				
					0x2		PCLK•1/4				
					0x1		PCLK•1/2				
					0x0		PCLK•1/1				
		D3–1	–	reserved	–				–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1	On	0	Off	0	R/W	

0x4200–0x4206 8-bit Timer (with Fine Mode)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
8-bit Timer Input Clock Select Register (T8F_CLK)	**0x4200** (16 bits)	D15–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			
8-bit Timer Reload Data Register (T8F_TR)	**0x4202** (16 bits)	D15–8	**–**	reserved	–		–	–	0 when being read.
		D7–0	**TR[7:0]**	8-bit timer reload data TR7 = MSB TR0 = LSB	0x0 to 0xff		0x0	R/W	
8-bit Timer Counter Data Register (T8F_TC)	**0x4204** (16 bits)	D15–8	**–**	reserved	–		–	–	0 when being read.
		D7–0	**TC[7:0]**	8-bit timer counter data TC7 = MSB TC0 = LSB	0x0 to 0xff		0xff	R	
8-bit Timer Control Register (T8F_CTL)	**0x4206** (16 bits)	D15–12	**–**	reserved	–		–	–	0 when being read.
		D11–8	**TFMD[3:0]**	Fine mode setup	0x0 to 0xf		0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	**–**	reserved	–		–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot	0 Repeat	0	R/W	
		D3–2	**–**	reserved	–		–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset	0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run	0 Stop	0	R/W	

0x4220–0x4246 16-bit Timer

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
16-bit Timer Ch.0 Input Clock Select Register (T16_CLK0)	**0x4220** (16 bits)	D15–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			
16-bit Timer Ch.0 Reload Data Register (T16_TR0)	**0x4222** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff		0x0	R/W	
16-bit Timer Ch.0 Counter Data Register (T16_TC0)	**0x4224** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff		0xffff	R	
16-bit Timer Ch.0 Control Register (T16_CTL0)	**0x4226** (16 bits)	D15–11	**–**	reserved	–		–	–	0 when being read.
		D10	**CKACTV**	External clock active level select	1 High	0 Low	1	R/W	
		D9–8	**CKSL[1:0]**	Input clock and pulse width measurement mode select	CKSL[1:0]	Mode	0x0	R/W	
					0x3	reserved			
					0x2	Pulse width			
					0x1	External clock			
					0x0	Internal clock			
		D7–5	**–**	reserved	–		–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot	0 Repeat	0	R/W	
		D3–2	**–**	reserved	–		–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset	0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run	0 Stop	0	R/W	
16-bit Timer Ch.1 Input Clock Select Register (T16_CLK1)	**0x4240** (16 bits)	D15–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			
16-bit Timer Ch.1 Reload Data Register (T16_TR1)	**0x4242** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff		0x0	R/W	
16-bit Timer Ch.1 Counter Data Register (T16_TC1)	**0x4244** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff		0xffff	R	
16-bit Timer Ch.1 Control Register (T16_CTL1)	**0x4246** (16 bits)	D15–11	**–**	reserved	–		–	–	0 when being read.
		D10	**CKACTV**	External clock active level select	1 High	0 Low	1	R/W	
		D9–8	**CKSL[1:0]**	Input clock and pulse width measurement mode select	CKSL[1:0]	Mode	0x0	R/W	
					0x3	reserved			
					0x2	Pulse width			
					0x1	External clock			
					0x0	Internal clock			
		D7–5	**–**	reserved	–		–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot	0 Repeat	0	R/W	
		D3–2	**–**	reserved	–		–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset	0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run	0 Stop	0	R/W	

0x4260–0x4266 16-bit Timer

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
16-bit Timer Ch.2 Input Clock Select Register (T16_CLK2)	**0x4260** (16 bits)	D15–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			
16-bit Timer Ch.2 Reload Data Register (T16_TR2)	**0x4262** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff		0x0	R/W	
16-bit Timer Ch.2 Counter Data Register (T16_TC2)	**0x4264** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff		0xffff	R	
16-bit Timer Ch.2 Control Register (T16_CTL2)	**0x4266** (16 bits)	D15–11	**–**	reserved	–		–	–	0 when being read.
		D10	**CKACTV**	External clock active level select	1 High	0 Low	1	R/W	
		D9–8	**CKSL[1:0]**	Input clock and pulse width measurement mode select	CKSL[1:0]	Mode	0x0	R/W	
					0x3	reserved			
					0x2	Pulse width			
					0x1	External clock			
					0x0	Internal clock			
		D7–5	**–**	reserved	–		–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot	0 Repeat	0	R/W	
		D3–2	**–**	reserved	–		–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset	0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run	0 Stop	0	R/W	

0x4300–0x430c Interrupt Controller

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Interrupt Flag Register (ITC_IFLG)	**0x4300** (16 bits)	D15	**IIFT7**	I²C interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D14	**IIFT6**	SPI interrupt flag					0	R/W	
		D13	**IIFT5**	Remote controller interrupt flag					0	R/W	
		D12	**IIFT4**	UART interrupt flag					0	R/W	
		D11	**IIFT3**	16-bit timer Ch.2 interrupt flag					0	R/W	
		D10	**IIFT2**	16-bit timer Ch.1 interrupt flag					0	R/W	
		D9	**IIFT1**	16-bit timer Ch.0 interrupt flag					0	R/W	
		D8	**IIFT0**	8-bit timer interrupt flag					0	R/W	
		D7	**EIFT7**	PWM&capture timer interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1 in pulse trigger mode. Cannot be reset by software in level trigger mode.
		D6	**EIFT6**	LCD interrupt flag					0	R/W	
		D5	**EIFT5**	SVD interrupt flag					0	R/W	
		D4	**EIFT4**	8-bit OSC1 timer interrupt flag					0	R/W	
		D3	**EIFT3**	Clock timer interrupt flag					0	R/W	
		D2	**EIFT2**	Stopwatch timer interrupt flag					0	R/W	
		D1	**EIFT1**	P1 port interrupt flag					0	R/W	
		D0	**EIFT0**	P0 port interrupt flag					0	R/W	
Interrupt Enable Register (ITC_EN)	**0x4302** (16 bits)	D15	**IIEN7**	I²C interrupt enable	1	Enable	0	Disable	0	R/W	
		D14	**IIEN6**	SPI interrupt enable					0	R/W	
		D13	**IIEN5**	Remote controller interrupt enable					0	R/W	
		D12	**IIEN4**	UART interrupt enable					0	R/W	
		D11	**IIEN3**	16-bit timer Ch.2 interrupt enable					0	R/W	
		D10	**IIEN2**	16-bit timer Ch.1 interrupt enable					0	R/W	
		D9	**IIEN1**	16-bit timer Ch.0 interrupt enable					0	R/W	
		D8	**IIEN0**	8-bit timer interrupt enable					0	R/W	
		D7	**EIEN7**	PWM&capture timer interrupt enable					0	R/W	
		D6	**EIEN6**	LCD interrupt enable					0	R/W	
		D5	**EIEN5**	SVD interrupt enable					0	R/W	
		D4	**EIEN4**	8-bit OSC1 timer interrupt enable					0	R/W	
		D3	**EIEN3**	Clock timer interrupt enable					0	R/W	
		D2	**EIEN2**	Stopwatch timer interrupt enable					0	R/W	
		D1	**EIEN1**	P1 port interrupt enable					0	R/W	
		D0	**EIEN0**	P0 port interrupt enable					0	R/W	
ITC Control Register (ITC_CTL)	**0x4304** (16 bits)	D15–1	–	reserved	–				–	–	0 when being read.
		D0	**ITEN**	ITC enable	1	Enable	0	Disable	0	R/W	
External Interrupt Level Setup Register 0 (ITC_ELV0)	**0x4306** (16 bits)	D15–13	–	reserved	–				–	–	0 when being read.
		D12	**EITG1**	P1 interrupt trigger mode	1	Level	0	Pulse	0	R/W	Be sure to set to 1.
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV1[2:0]**	P1 interrupt level	0 to 7				0x0	R/W	
		D7–5	–	reserved	–				–	–	0 when being read.
		D4	**EITG0**	P0 interrupt trigger mode	1	Level	0	Pulse	0	R/W	Be sure to set to 1.
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV0[2:0]**	P0 interrupt level	0 to 7				0x0	R/W	
External Interrupt Level Setup Register 1 (ITC_ELV1)	**0x4308** (16 bits)	D15–13	–	reserved	–				–	–	0 when being read.
		D12	**EITG3**	CT interrupt trigger mode	1	Level	0	Pulse	0	R/W	Be sure to set to 1.
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV3[2:0]**	CT interrupt level	0 to 7				0x0	R/W	
		D7–5	–	reserved	–				–	–	0 when being read.
		D4	**EITG2**	SWT interrupt trigger mode	1	Level	0	Pulse	0	R/W	Be sure to set to 1.
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV2[2:0]**	SWT interrupt level	0 to 7				0x0	R/W	
External Interrupt Level Setup Register 2 (ITC_ELV2)	**0x430a** (16 bits)	D15–13	–	reserved	–				–	–	0 when being read.
		D12	**EITG5**	SVD interrupt trigger mode	1	Level	0	Pulse	0	R/W	Be sure to set to 1.
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV5[2:0]**	SVD interrupt level	0 to 7				0x0	R/W	
		D7–5	–	reserved	–				–	–	0 when being read.
		D4	**EITG4**	T8OSC1 interrupt trigger mode	1	Level	0	Pulse	0	R/W	Be sure to set to 1.
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV4[2:0]**	T8OSC1 interrupt level	0 to 7				0x0	R/W	
External Interrupt Level Setup Register 3 (ITC_ELV3)	**0x430c** (16 bits)	D15–13	–	reserved	–				–	–	0 when being read.
		D12	**EITG7**	T16E interrupt trigger mode	1	Level	0	Pulse	0	R/W	Be sure to set to 1.
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV7[2:0]**	T16E interrupt level	0 to 7				0x0	R/W	
		D7–5	–	reserved	–				–	–	0 when being read.
		D4	**EITG6**	LCD interrupt trigger mode	1	Level	0	Pulse	0	R/W	Be sure to set to 1.
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV6[2:0]**	LCD interrupt level	0 to 7				0x0	R/W	

0x430e–0x4314 Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 0 (ITC_ILV0)	**0x430e** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV1[2:0]**	T16 Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV0[2:0]**	T8 interrupt level	0 to 7	0x0	R/W	
Internal Interrupt Level Setup Register 1 (ITC_ILV1)	**0x4310** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV3[2:0]**	T16 Ch.2 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV2[2:0]**	T16 Ch.1 interrupt level	0 to 7	0x0	R/W	
Internal Interrupt Level Setup Register 2 (ITC_ILV2)	**0x4312** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV5[2:0]**	REMC interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV4[2:0]**	UART interrupt level	0 to 7	0x0	R/W	
Internal Interrupt Level Setup Register 3 (ITC_ILV3)	**0x4314** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV7[2:0]**	I^{2}C interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV6[2:0]**	SPI interrupt level	0 to 7	0x0	R/W	

0x4320–0x4326 SPI

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Status Register (SPI_ST)	**0x4320** (16 bits)	D15–3	**–**	reserved	–				–	–	0 when being read.
		D2	**SPBSY**	Transfer busy flag (master)	1	Busy	0	Idle	0	R	
				ss signal low flag (slave)	1	ss = L	0	ss = H			
		D1	**SPRBF**	Receive data buffer full flag	1	Full	0	Not full	0	R	
		D0	**SPTBE**	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	
SPI Transmit Data Register (SPI_TXD)	**0x4322** (16 bits)	D15–8	**–**	reserved	–				–	–	0 when being read.
		D7–0	**SPTDB[7:0]**	SPI transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff				0x0	R/W	
SPI Receive Data Register (SPI_RXD)	**0x4324** (16 bits)	D15–8	**–**	reserved	–				–	–	0 when being read.
		D7–0	**SPRDB[7:0]**	SPI receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff				0x0	R	
SPI Control Register (SPI_CTL)	**0x4326** (16 bits)	D15–6	**–**	reserved	–				–	–	0 when being read.
		D5	**SPRIE**	Receive data buffer full int. enable	1	Enable	0	Disable	0	R/W	
		D4	**SPTIE**	Transmit data buffer empty int. enable	1	Enable	0	Disable	0	R/W	
		D3	**CPHA**	Clock phase select	1	Data out	0	Data in	0	R/W	These bits must be set before setting SPEN to 1.
		D2	**CPOL**	Clock polarity select	1	Active L	0	Active H	0	R/W	
		D1	**MSSL**	Master/slave mode select	1	Master	0	Slave	0	R/W	
		D0	**SPEN**	SPI enable	1	Enable	0	Disable	0	R/W	

0x4340–0x4346 I^{2}C

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^{2}C Enable Register (I2C_EN)	**0x4340** (16 bits)	D15–1	**–**	reserved	–				–	–	0 when being read.
		D0	**I2CEN**	I^{2}C enable	1	Enable	0	Disable	0	R/W	
I^{2}C Control Register (I2C_CTL)	**0x4342** (16 bits)	D15–10	**–**	reserved	–				–	–	0 when being read.
		D9	**RBUSY**	Receive busy flag	1	Busy	0	Idle	0	R	
		D8	**TBUSY**	Transmit busy flag	1	Busy	0	Idle	0	R	
		D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**NSERM**	Noise remove on/off	1	On	0	Off	0	R/W	
		D3–2	**–**	reserved	–				–	–	0 when being read.
		D1	**STP**	Stop control	1	Stop	0	Ignored	0	R/W	
		D0	**STRT**	Start control	1	Start	0	Ignored	0	R/W	
I^{2}C Data Register (I2C_DAT)	**0x4344** (16 bits)	D15–12	**–**	reserved	–				–	–	0 when being read.
		D11	**RBRDY**	Receive buffer ready	1	Ready	0	Empty	0	R	
		D10	**RXE**	Receive execution	1	Receive	0	Ignored	0	R/W	
		D9	**TXE**	Transmit execution	1	Transmit	0	Ignored	0	R/W	
		D8	**RTACK**	Receive/transmit ACK	1	Error	0	ACK	0	R/W	
		D7–0	**RTDT[7:0]**	Receive/transmit data RTDT7 = MSB RTDT0 = LSB	0x0 to 0xff				0x0	R/W	
I^{2}C Interrupt Control Register (I2C_ICTL)	**0x4346** (16 bits)	D15–2	**–**	reserved	–				–	–	0 when being read.
		D1	**RINTE**	Receive interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**TINTE**	Transmit interrupt enable	1	Enable	0	Disable	0	R/W	

0x5000–0x5003 Clock Timer

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Clock Timer Control Register (CT_CTL)	**0x5000** (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**CTRST**	Clock timer reset	1	Reset	0	Ignored	0	W	
		D3–1	**–**	reserved	–				–	–	
		D0	**CTRUN**	Clock timer run/stop control	1	Run	0	Stop	0	R/W	
Clock Timer Counter Register (CT_CNT)	**0x5001** (8 bits)	D7–0	**CTCNT[7:0]**	Clock timer counter value	0x0 to 0xff				0	R	
Clock Timer Interrupt Mask Register (CT_IMSK)	**0x5002** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CTIE32**	32 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D2	**CTIE8**	8 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**CTIE2**	2 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**CTIE1**	1 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
Clock Timer Interrupt Flag Register (CT_IFLG)	**0x5003** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CTIF32**	32 Hz interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D2	**CTIF8**	8 Hz interrupt flag					0	R/W	
		D1	**CTIF2**	2 Hz interrupt flag					0	R/W	
		D0	**CTIF1**	1 Hz interrupt flag					0	R/W	

0x5020–0x5023 Stopwatch Timer

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Stopwatch Timer Control Register (SWT_CTL)	**0x5020** (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**SWTRST**	Stopwatch timer reset	1	Reset	0	Ignored	0	W	
		D3–1	**–**	reserved	–				–	–	
		D0	**SWTRUN**	Stopwatch timer run/stop control	1	Run	0	Stop	0	R/W	
Stopwatch Timer BCD Counter Register (SWT_BCNT)	**0x5021** (8 bits)	D7–4	**BCD10[3:0]**	1/10 sec. BCD counter value	0 to 9				0	R	
		D3–0	**BCD100[3:0]**	1/100 sec. BCD counter value	0 to 9				0	R	
Stopwatch Timer Interrupt Mask Register (SWT_IMSK)	**0x5022** (8 bits)	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**SIE1**	1 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**SIE10**	10 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**SIE100**	100 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
Stopwatch Timer Interrupt Flag Register (SWT_IFLG)	**0x5023** (8 bits)	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**SIF1**	1 Hz interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	**SIF10**	10 Hz interrupt flag					0	R/W	
		D0	**SIF100**	100 Hz interrupt flag					0	R/W	

0x5040–0x5041 Watchdog Timer

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Watchdog Timer Control Register (WDT_CTL)	**0x5040** (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**WDTRST**	Watchdog timer reset	1	Reset	0	Ignored	0	W	
		D3–0	**WDTRUN[3:0]**	Watchdog timer run/stop control	Other than 1010 Run		1010 Stop		1010	R/W	
Watchdog Timer Status Register (WDT_ST)	**0x5041** (8 bits)	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**WDTMD**	NMI/Reset mode select	1	Reset	0	NMI	0	R/W	
		D0	**WDTST**	NMI status	1	NMI occurred	0	Not occurred	0	R	

0x5060–0x5065 Oscillator

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Clock Source Select Register (OSC_SRC)	**0x5060** (8 bits)	D7–1	**–**	reserved	–		–	–	0 when being read.
		D0	**CLKSRC**	System clock source select	1 OSC1	0 OSC3	0	R/W	
Oscillation Control Register (OSC_CTL)	**0x5061** (8 bits)	D7–6	**–**	reserved	–		–	–	0 when being read.
		D5–4	**OSC3WT[1:0]**	OSC3 wait cycle select	OSC3WT[1:0]	Wait cycle	0x0	R/W	
					0x3	128 cycles			
					0x2	256 cycles			
					0x1	512 cycles			
					0x0	1024 cycles			
		D3–2	**–**	reserved	–		–	–	0 when being read.
		D1	**OSC1EN**	OSC1 enable	1 Enable	0 Disable	1	R/W	
		D0	**OSC3EN**	OSC3 enable	1 Enable	0 Disable	1	R/W	
Noise Filter Enable Register (OSC_NFEN)	**0x5062** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1	**RSTFE**	Reset noise filter enable	1 Enable	0 Disable	1	R/W	
		D0	**NMIFE**	NMI noise filter enable	1 Enable	0 Disable	0	R/W	
LCD Clock Setup Register (OSC_LCLK)	**0x5063** (8 bits)	D7–5	**–**	reserved	–		–	–	0 when being read.
		D4–2	**LCKDV[2:0]**	LCD clock division ratio select	LCKDV[2:0]	Division ratio	0x0	R/W	
					0x7–0x5	reserved			
					0x4	OSC3•1/512			
					0x3	OSC3•1/256			
					0x2	OSC3•1/128			
					0x1	OSC3•1/64			
					0x0	OSC3•1/32			
		D1	**LCKSRC**	LCD clock source select	1 OSC1	0 OSC3	1	R/W	
		D0	**LCKEN**	LCD clock enable	1 Enable	0 Disable	0	R/W	
FOUT Control Register (OSC_FOUT)	**0x5064** (8 bits)	D7–4	**–**	reserved	–		–	–	0 when being read.
		D3–2	**FOUT3D[1:0]**	FOUT3 clock division ratio select	FOUT3D[1:0]	Division ratio	0x0	R/W	
					0x3	reserved			
					0x2	OSC3•1/4			
					0x1	OSC3•1/2			
					0x0	OSC3•1/1			
		D1	**FOUT3E**	FOUT3 output enable	1 Enable	0 Disable	0	R/W	
		D0	**FOUT1E**	FOUT1 output enable	1 Enable	0 Disable	0	R/W	
T8OSC1 Clock Control Register (OSC_T8OSC1)	**0x5065** (8 bits)	D7–4	**–**	reserved	–		–	–	0 when being read.
		D3–1	**T8O1CK[2:0]**	T8OSC1 clock division ratio select	T8O1CK[2:0]	Division ratio	0x0	R/W	
					0x7–0x6	reserved			
					0x5	OSC1•1/32			
					0x4	OSC1•1/16			
					0x3	OSC1•1/8			
					0x2	OSC1•1/4			
					0x1	OSC1•1/2			
					0x0	OSC1•1/1			
		D0	**T8O1CE**	T8OSC1 clock output enable	1 Enable	0 Disable	0	R/W	

0x5080–0x5081 Clock Generator

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
PCLK Control Register (CLG_PCLK)	**0x5080** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**PCKEN[1:0]**	PCLK enable	PCKEN[1:0]	PCLK supply	0x3	R/W	
					0x3	Enable			
					0x2	Not allowed			
					0x1	Not allowed			
					0x0	Disable			
CCLK Control Register (CLG_CCLK)	**0x5081** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**CCLKGR[1:0]**	CCLK clock gear ratio select	CCLKGR[1:0]	Gear ratio	0x0	R/W	
					0x3	1/8			
					0x2	1/4			
					0x1	1/2			
					0x0	1/1			

0x50a0–0x50a6 LCD Driver

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
LCD Display Control Register (LCD_DCTL)	**0x50a0** (8 bits)	D7	**SEGREV**	Segment output assignment control	1 Normal	0 Reverse	1	R/W	
		D6	**COMREV**	Common output assignment control	1 Normal	0 Reverse	1	R/W	
		D5	**DSPAR**	Display memory area control	1 Area 1	0 Area 0	0	R/W	
		D4	**DSPREV**	Reverse display control	1 Normal	0 Reverse	1	R/W	
		D3–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**DSPC[1:0]**	LCD display control	DSPC[1:0]	Display	0x0	R/W	
					0x3 0x2 0x1 0x0	All off All on Normal display Display off			
LCD Contrast Adjust Register (LCD_CADJ)	**0x50a1** (8 bits)	D7–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**LC[3:0]**	LCD contrast adjustment	LC[3:0]	Display	0x0	R/W	
					0xf : 0x0	Dark : Light			
LCD Clock Control Register (LCD_CCTL)	**0x50a2** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**LDUTY[1:0]**	LCD duty select	LDUTY[1:0]	Duty	0x2	R/W	
					0x3 0x2 0x1 0x0	reserved 1/32 1/16 reserved			
LCD Voltage Regulator Control Register (LCD_VREG)	**0x50a3** (8 bits)	D7–5	**–**	reserved	–		–	–	0 when being read.
		D4	**LHVLD**	LCD heavy load protection mode	1 On	0 Off	0	R/W	
		D3–0	**–**	reserved	–		–	–	0 when being read.
LCD Power Voltage Booster Control Register (LCD_PWR)	**0x50a4** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1	**VDSEL**	Regulator power source select	1 V_{D2}	0 V_{DD}	0	R/W	
		D0	**PBON**	Power voltage booster control	1 On	0 Off	0	R/W	
LCD Interrupt Mask Register (LCD_IMSK)	**0x50a5** (8 bits)	D7–1	**–**	reserved	–		–	–	0 when being read.
		D0	**FRMIE**	Frame signal interrupt enable	1 Enable	0 Disable	0	R/W	
LCD Interrupt Flag Register (LCD_IFLG)	**0x50a6** (8 bits)	D7–1	**–**	reserved	–		–	–	0 when being read.
		D0	**FRMIF**	Frame signal interrupt flag	1 Cccurred	0 Not occurred	0	R/W	Reset by writing 1.

0x50c0–0x50c4 8-bit OSC1 Timer

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1 Timer Control Register (T8OSC1_CTL)	**0x50c0** (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**T8ORST**	Timer reset	1	Reset	0	Ignored	0	W	
		D3–2	**–**	reserved	–				–	–	
		D1	**T8ORMD**	Count mode select	1	One shot	0	Repeat	0	R/W	
		D0	**T8ORUN**	Timer run/stop control	1	Run	0	Stop	0	R/W	
8-bit OSC1 Timer Counter Data Register (T8OSC1_CNT)	**0x50c1** (8 bits)	D7–0	**T8OCNT[7:0]**	Timer counter data T8OCNT7 = MSB T8OCNT0 = LSB	0x0 to 0xff				0x0	R	
8-bit OSC1 Timer Compare Data Register (T8OSC1_CMP)	**0x50c2** (8 bits)	D7–0	**T8OCMP[7:0]**	Compare data T8OCMP7 = MSB T8OCMP0 = LSB	0x0 to 0xff				0x0	R/W	
8-bit OSC1 Timer Interrupt Mask Register (T8OSC1_IMSK)	**0x50c3** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**T8OIE**	8-bit OSC1 timer interrupt enable	1	Enable	0	Disable	0	R/W	
8-bit OSC1 Timer Interrupt Flag Register (T8OSC1_IFLG)	**0x50c4** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**T8OIF**	8-bit OSC1 timer interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x5100–0x5104 SVD Circuit

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SVD Enable Register (SVD_EN)	**0x5100** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**SVDEN**	SVD enable	1	Enable	0	Disable	0	R/W	
SVD Compare Voltage Register (SVD_CMP)	**0x5101** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3–0	**SVDC[3:0]**	SVD compare voltage	SVDC[3:0]		Voltage		0x0	R/W	
					0xf		2.7 V				
					0xe		2.6 V				
					0xd		2.5 V				
					0xc		2.4 V				
					0xb		2.3 V				
					0xa		2.2 V				
					0x9		2.1 V				
					0x8		2.05 V				
					0x7		2.0 V				
					0x6		1.95 V				
					0x5		1.9 V				
					0x4		1.85 V				
					0x3		1.8 V				
					0x2		–				
					0x1		–				
					0x0		–				
SVD Detection Result Register (SVD_RSLT)	**0x5102** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**SVDDT**	SVD detection result	1	Low	0	Normal	×	R	
SVD Interrupt Mask Register (SVD_IMSK)	**0x5103** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**SVDIE**	SVD interrupt enable	1	Enable	0	Disable	0	R/W	
SVD Interrupt Flag Register (SVD_IFLG)	**0x5104** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**SVDIF**	SVD interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x5120 Power Generator

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
V_{D1} Control Register (VD1_CTL)	**0x5120** (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**HVLD**	V_{D1} heavy load protection mode	1	On	0	Off	0	R/W	
		D3–1	**–**	reserved	–				–	–	0 when being read.
		D0	**VD1MD**	Flash erase/program mode	1	Flash (2.5 V)	0	Norm.(1.8 V)	0	R/W	

0x5200–0x5214 P Port & Port MUX

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Input Data Register (P0_IN)	**0x5200** (8 bits)	D7–0	**P0IN[7:0]**	P0[7:0] port input data	1	1 (H)	0	0 (L)	×	R	
P0 Port Output Data Register (P0_OUT)	**0x5201** (8 bits)	D7–0	**P0OUT[7:0]**	P0[7:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P0 Port I/O Direction Control Register (P0_IO)	**0x5202** (8 bits)	D7–0	**P0IO[7:0]**	P0[7:0] port I/O direction select	1	Output	0	Input	0	R/W	
P0 Port Pull-up Control Register (P0_PU)	**0x5203** (8 bits)	D7–0	**P0PU[7:0]**	P0[7:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	
P0 Port Schmitt Trigger Control Register (P0_SM)	**0x5204** (8 bits)	D7–0	**P0SM[7:0]**	P0[7:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1 (0xff)	R/W	
P0 Port Interrupt Mask Register (P0_IMSK)	**0x5205** (8 bits)	D7–0	**P0IE[7:0]**	P0[7:0] port interrupt enable	1	Enable	0	Disable	0	R/W	
P0 Port Interrupt Edge Select Register (P0_EDGE)	**0x5206** (8 bits)	D7–0	**P0EDGE[7:0]**	P0[7:0] port interrupt edge select	1	Falling edge	0	Rising edge	0	R/W	
P0 Port Interrupt Flag Register (P0_IFLG)	**0x5207** (8 bits)	D7–0	**P0IF[7:0]**	P0[7:0] port interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P0 Port Chattering Filter Control Register (P0_CHAT)	**0x5208** (8 bits)	D7	**–**	reserved	–				–	–	0 when being read.
		D6–4	**P0CF2[2:0]**	P0[7:4] chattering filter time	P0CF2[2:0]		Filter time		0	R/W	
					0x7 0x6 0x5 0x4 0x3 0x2 0x1 0x0		16384/fPCLK 8192/fPCLK 4096/fPCLK 2048/fPCLK 1024/fPCLK 512/fPCLK 256/fPCLK None		0x0	R/W	
		D3	**–**	reserved	–				–	–	0 when being read.
		D2–0	**P0CF1[2:0]**	P0[3:0] chattering filter time	P0CF1[2:0]		Filter time		0x0	R/W	
					0x7 0x6 0x5 0x4 0x3 0x2 0x1 0x0		16384/fPCLK 8192/fPCLK 4096/fPCLK 2048/fPCLK 1024/fPCLK 512/fPCLK 256/fPCLK None				
P0 Port Key-Entry Reset Configuration Register (P0_KRST)	**0x5209** (8 bits)	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1–0	**P0KRST[1:0]**	P0 port key-entry reset configuration	P0KRST[1:0]		Configuration		0x0	R/W	
					0x3 0x2 0x1 0x0		P0[3:0] = 0 P0[2:0] = 0 P0[1:0] = 0 Disable				
P1 Port Input Data Register (P1_IN)	**0x5210** (8 bits)	D7–0	**P1IN[7:0]**	P1[7:0] port input data	1	1 (H)	0	0 (L)	×	R	
P1 Port Output Data Register (P1_OUT)	**0x5211** (8 bits)	D7–0	**P1OUT[7:0]**	P1[7:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P1 Port I/O Direction Control Register (P1_IO)	**0x5212** (8 bits)	D7–0	**P1IO[7:0]**	P1[7:0] port I/O direction select	1	Output	0	Input	0	R/W	
P1 Port Pull-up Control Register (P1_PU)	**0x5213** (8 bits)	D7–0	**P1PU[7:0]**	P1[7:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	
P1 Port Schmitt Trigger Control Register (P1_SM)	**0x5214** (8 bits)	D7–0	**P1SM[7:0]**	P1[7:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1 (0xff)	R/W	

0x5215–0x52a3 P Port & Port MUX

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P1 Port Interrupt Mask Register (P1_IMSK)	**0x5215** (8 bits)	D7–0	**P1IE[7:0]**	P1[7:0] port interrupt enable	1	Enable	0	Disable	0	R/W	
P1 Port Interrupt Edge Select Register (P1_EDGE)	**0x5216** (8 bits)	D7–0	**P1EDGE[7:0]**	P1[7:0] port interrupt edge select	1	Falling edge	0	Rising edge	0	R/W	
P1 Port Interrupt Flag Register (P1_IFLG)	**0x5217** (8 bits)	D7–0	**P1IF[7:0]**	P1[7:0] port interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P2 Port Input Data Register (P2_IN)	**0x5220** (8 bits)	D7–0	**P2IN[7:0]**	P2[7:0] port input data	1	1 (H)	0	0 (L)	×	R	
P2 Port Output Data Register (P2_OUT)	**0x5221** (8 bits)	D7–0	**P2OUT[7:0]**	P2[7:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P2 Port I/O Direction Control Register (P2_IO)	**0x5222** (8 bits)	D7–0	**P2IO[7:0]**	P2[7:0] port I/O direction select	1	Output	0	Input	0	R/W	
P2 Port Pull-up Control Register (P2_PU)	**0x5223** (8 bits)	D7–0	**P2PU[7:0]**	P2[7:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	
P2 Port Schmitt Trigger Control Register (P2_SM)	**0x5224** (8 bits)	D7–0	**P2SM[7:0]**	P2[7:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1 (0xff)	R/W	
P3 Port Input Data Register (P3_IN)	**0x5230** (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	**P3IN[3:0]**	P3[3:0] port input data	1	1 (H)	0	0 (L)	×	R	
P3 Port Output Data Register (P3_OUT)	**0x5231** (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	**P3OUT[3:0]**	P3[3:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P3 Port I/O Direction Control Register (P3_IO)	**0x5232** (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	**P3IO[3:0]**	P3[3:0] port I/O direction select	1	Output	0	Input	0	R/W	
P3 Port Pull-up Control Register (P3_PU)	**0x5233** (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	**P3PU[3:0]**	P3[3:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	
P3 Port Schmitt Trigger Control Register (P3_SM)	**0x5234** (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	**P3SM[3:0]**	P3[3:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1 (0xff)	R/W	
P0 Port Function Select Register (P0_PMUX)	**0x52a0** (8 bits)	D7–6	–	reserved	–				–	–	0 when being read.
		D5	**P05MUX**	P05 port function select	1	REMO	0	P05	0	R/W	
		D4	**P04MUX**	P04 port function select	1	REMI	0	P04	0	R/W	
		D3–0	–	reserved	–				–	–	0 when being read.
P1 Port Function Select Register (P1_PMUX)	**0x52a1** (8 bits)	D7	**P17MUX**	P17 port function select	1	#SPISS	0	P17	0	R/W	
		D6	–	reserved	–				–	–	0 when being read.
		D5	**P15MUX**	P15 port function select	1	SCL	0	P15	0	R/W	
		D4	**P14MUX**	P14 port function select	1	SDA	0	P14	0	R/W	
		D3	**P13MUX**	P13 port function select	1	FOUT1	0	P13	0	R/W	
		D2–0	–	reserved	–				–	–	0 when being read.
P2 Port Function Select Register (P2_PMUX)	**0x52a2** (8 bits)	D7	**P27MUX**	P27 port function select	1	EXCL3	0	P27	0	R/W	
		D6	**P26MUX**	P26 port function select	1	TOUT	0	P26	0	R/W	
		D5	**P25MUX**	P25 port function select	1	SCLK	0	P25	0	R/W	
		D4	**P24MUX**	P24 port function select	1	SOUT	0	P24	0	R/W	
		D3	**P23MUX**	P23 port function select	1	SIN	0	P23	0	R/W	
		D2	**P22MUX**	P22 port function select	1	SPICLK	0	P22	0	R/W	
		D1	**P21MUX**	P21 port function select	1	SDO	0	P21	0	R/W	
		D0	**P20MUX**	P20 port function select	1	SDI	0	P20	0	R/W	
P3 Port Function Select Register (P3_PMUX)	**0x52a3** (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**P33MUX**	P33 port function select	1	P33	0	DSIO	0	R/W	
		D2	**P32MUX**	P32 port function select	1	P32	0	DST2	0	R/W	
		D1	**P31MUX**	P31 port function select	1	P31	0	DCLK	0	R/W	
		D0	**P30MUX**	P30 port function select	1	FOUT3	0	P30	0	R/W	

0x5300–0x530c PWM & Capture Timer

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
PWM Timer Compare Data A Register (T16E_CA)	**0x5300** (16 bits)	D15–0	**T16ECA[15:0]**	Compare data A T16ECA15 = MSB T16ECA0 = LSB	0x0 to 0xffff				0x0	R/W	
PWM Timer Compare Data B Register (T16E_CB)	**0x5302** (16 bits)	D15–0	**T16ECB[15:0]**	Compare data B T16ECB15 = MSB T16ECB0 = LSB	0x0 to 0xffff				0x0	R/W	
PWM Timer Counter Data Register (T16E_TC)	**0x5304** (16 bits)	D15–0	**T16ETC[15:0]**	Counter data T16ETC15 = MSB T16ETC0 = LSB	0x0 to 0xffff				0x0	R/W	
PWM Timer Control Register (T16E_CTL)	**0x5306** (16 bits)	D15–9	**–**	reserved	–				–	–	0 when being read.
		D8	**INITOL**	Initial output level	1	High	0	Low	0	R/W	
		D7	**–**	reserved	–				–	–	0 when being read.
		D6	**SELFM**	Fine mode select	1	Fine mode	0	Normal mode	0	R/W	
		D5	**CBUFEN**	Comparison buffer enable	1	Enable	0	Disable	0	R/W	
		D4	**INVOUT**	Inverse output	1	Invert	0	Normal	0	R/W	
		D3	**CLKSEL**	Input clock select	1	External	0	Internal	0	R/W	
		D2	**OUTEN**	Clock output enable	1	Enable	0	Disable	0	R/W	
		D1	**T16ERST**	Timer reset	1	Reset	0	Ignored	0	W	0 when being read.
		D0	**T16ERUN**	Timer run/stop control	1	Run	0	Stop	0	R/W	
PWM Timer Input Clock Select Register (T16E_CLK)	**0x5308** (16 bits)	D15–4	**–**	reserved	–				–	–	0 when being read.
		D3–0	**T16EDF[3:0]**	Timer input clock select (Prescaler output clock)	T16EDF[3:0]	Clock			0x0	R/W	
					0xf	reserved					
					0xe	PCLK•1/16384					
					0xd	PCLK•1/8192					
					0xc	PCLK•1/4096					
					0xb	PCLK•1/2048					
					0xa	PCLK•1/1024					
					0x9	PCLK•1/512					
					0x8	PCLK•1/256					
					0x7	PCLK•1/128					
					0x6	PCLK•1/64					
					0x5	PCLK•1/32					
					0x4	PCLK•1/16					
					0x3	PCLK•1/8					
					0x2	PCLK•1/4					
					0x1	PCLK•1/2					
					0x0	PCLK•1/1					
PWM Timer Interrupt Mask Register (T16E_IMSK)	**0x530a** (16 bits)	D15–2	**–**	reserved	–				–	–	0 when being read.
		D1	**CBIE**	Compare B interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**CAIE**	Compare A interrupt enable	1	Enable	0	Disable	0	R/W	
PWM Timer Interrupt Flag Register (T16E_IFLG)	**0x530c** (16 bits)	D15–2	**–**	reserved	–				–	–	0 when being read.
		D1	**CBIF**	Compare B interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D0	**CAIF**	Compare A interrupt flag					0	R/W	

0x5320–0x5322 MISC Registers

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
FLASHC Control Register (MISC_FL)	**0x5320**	D7–3	**–**	reserved	–		–	–	0 when being read.
	(8 bits)	D2–0	**FLCYC[2:0]**	FLASHC read access cycle	FLCYC[2:0]	Read cycle	0x3	R/W	
					0x7–0x5	reserved			
					0x4	1.5 cycles			
					0x3	5.5 cycles			
					0x2	4.5 cycles			
					0x1	3.5 cycles			
					0x0	2.5 cycles			
SRAMC Control Register (MISC_SR)	**0x5321**	D7–2	**–**	reserved	–		–	–	0 when being read.
	(8 bits)	D1–0	**SRCYC[1:0]**	SRAMC access cycle	SRWAIT[1:0]	Access cycle	0x3	R/W	
					0x3	5 cycles			
					0x2	4 cycles			
					0x1	3 cycles			
					0x0	2 cycles			
OSC1 Peripheral Control Register (MISC_OSC1)	**0x5322**	D7–1	**–**	reserved	–		–	–	0 when being read.
	(8 bits)	D0	**O1DBG**	OSC1 peripheral control in debug mode	1 Run	0 Stop	0	R/W	

0x5340–0x5347 Remote Controller

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
REMC Configuration Register (REMC_CFG)	**0x5340** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1	**REMMD**	REMC mode select	1 Receive	0 Transmit	0	R/W	
		D0	**REMEN**	REMC enable	1 Enable	0 Disable	0	R/W	
REMC Prescaler Clock Select Register (REMC_PSC)	**0x5341** (8 bits)	D7–4	**CGCLK[3:0]**	Carrier generator clock select (Prescaler output clock)	CGCLK[3:0] LCCLK[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
		D3–0	**LCCLK[3:0]**	Length counter clock select (Prescaler output clock)	0x8	PCLK•1/256	0x0	R/W	
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			
REMC H Carrier Length Setup Register (REMC_CARH)	**0x5342** (8 bits)	D7–6	**–**	reserved	–		–	–	0 when being read.
		D5–0	**REMCH[5:0]**	H carrier length setup	0x0 to 0x3f		0x0	R/W	
REMC L Carrier Length Setup Register (REMC_CARL)	**0x5343** (8 bits)	D7–6	**–**	reserved	–		–	–	0 when being read.
		D5–0	**REMCL[5:0]**	L carrier length setup	0x0 to 0x3f		0x0	R/W	
REMC Status Register (REMC_ST)	**0x5344** (8 bits)	D7–1	**–**	reserved	–		–	–	0 when being read.
		D0	**REMDT**	Transmit/receive data	1 1 (H)	0 0 (L)	0	R/W	
REMC Length Counter Register (REMC_LCNT)	**0x5345** (8 bits)	D7–0	**REMLEN[7:0]**	Transmit/receive data length count (down counter)	0x0 to 0xff		0x0	R/W	
REMC Interrupt Mask Register (REMC_IMSK)	**0x5346** (8 bits)	D7–3	**–**	reserved	–		–	–	0 when being read.
		D2	**REMFIE**	Falling edge interrupt enable	1 Enable	0 Disable	0	R/W	
		D1	**REMRIE**	Rising edge interrupt enable	1 Enable	0 Disable	0	R/W	
		D0	**REMUIE**	Underflow interrupt enable	1 Enable	0 Disable	0	R/W	
REMC Interrupt Flag Register (REMC_IFLG)	**0x5347** (8 bits)	D7–3	**–**	reserved	–		–	–	0 when being read.
		D2	**REMFIF**	Falling edge interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	**REMRIF**	Rising edge interrupt flag			0	R/W	
		D0	**REMUIF**	Underflow interrupt flag			0	R/W	

0xffff80–0xffff90 S1C17 Core I/O

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Vector Table Base Register (TTBR)	**0xffff80** (32 bits)	D31–24	–	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	**TTBR[23:0]**	Vector table base address	0x8000	0x80 00	R	
Processor ID Register (IDIR)	**0xffff84** (8 bits)	D7–0	**IDIR[7:0]**	Processor ID 0x10: S1C17 Core	0x10	0x10	R	
Debug RAM Base Register (DBRAM)	**0xffff90** (32 bits)	D31–24	–	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	**DBRAM[23:0]**	Debug RAM base address	0xfc0	0xfc0	R	

THIS PAGE IS BLANK.

Appendix B Flash Programming

There are two Flash memory programming methods available, programming from the debugger with an ICD (In-Circuit Debugger) such as S5U1C17001H (ICD Mini) that contains a Flash programmer function and self-programming controlled by the application program.

B.1 Programming from Debugger

The debugger included in the S1C17 Family C Compiler Package supports a Flash programmer function using an ICD (e.g. S5U1C17001H).

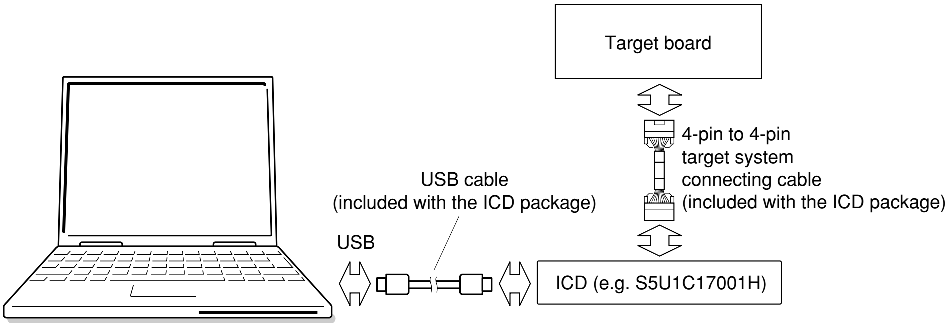


Figure B.1.1 Flash Programming System using Debugger

To program the S1C17701 Flash memory using this function, a four-pin connector is required on the target board for connecting the ICD (e.g. S5U1C17001H).
Use the S1C17701 DCLK (P31), DST2 (P32), and DSIO (P33) pins as the debug pins and connect them to the four-pin connector. In this case, the P31 to P33 general-purpose I/O ports cannot be used.

For the Flash programming procedures using this system and the pin assignment of the four-pin connector, refer to the manuals included in the S1C17 Family C compiler package (e.g. S5U1C17001C) and ICD package (e.g. S5U1C17001H), respectively.

B.2 Self-Programming by Application Program

The S1C17701 has a self-programming function that allows the application program being executed to erase and program the Flash memory while the S1C17701 is running on the target board.

For the S1C17701, an object file that includes the functional routines for self-programming is provided as the self-programming package.

By linking this object with the application program, a self-programming function can be implemented easily. For details, refer to the manual supplied with the self-programming package.

Appendix C Power Saving

Current consumption depends, to a large degree, on the CPU operating mode, operating clock frequency, and the peripheral circuits to be activated. This chapter summarizes the control to save power.

C.1 Power Saving by Clock Control

Figure C.1.1 shows the S1C17701 clock system.

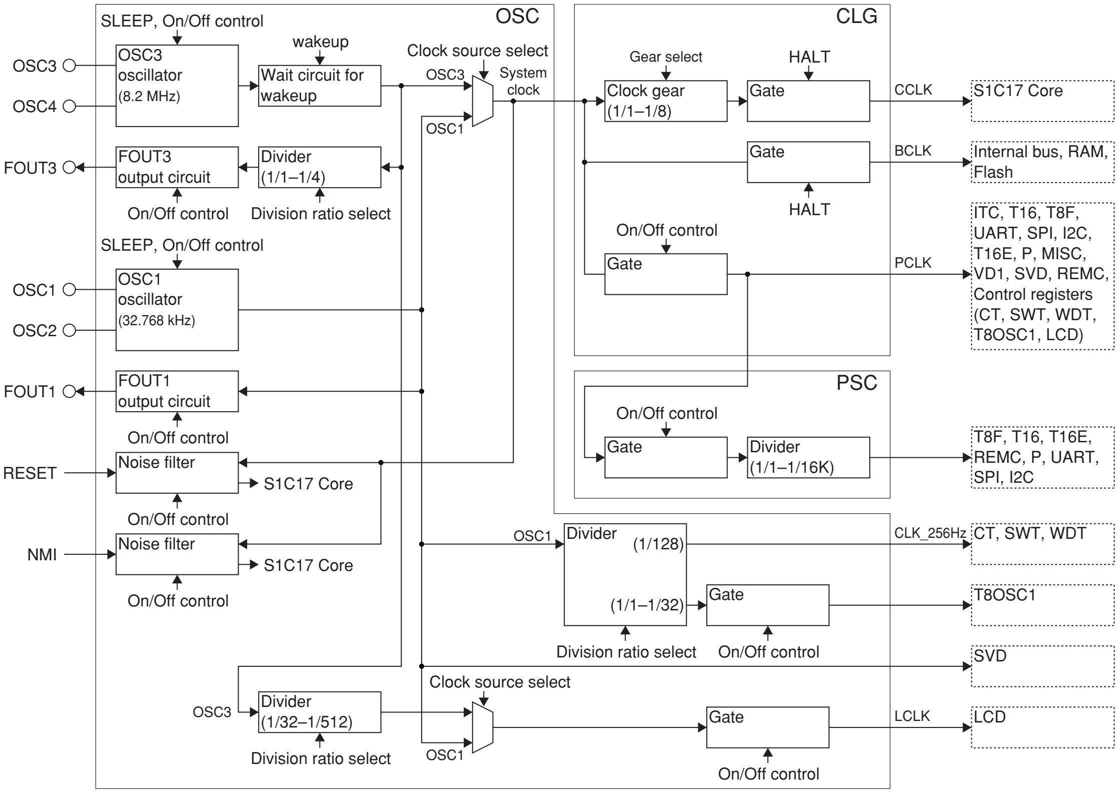


Figure C.1.1 Clock System

The following shows the clock systems that can be controlled with software and power saving control methods. For details of control registers and control methods, see the chapter for each module.

System sleep (disabling all clocks)

- Executing the `slp` instruction
  Execute the `slp` instruction if all of the system can be stopped. The CPU enters SLEEP mode and the OSC1 and OSC3 oscillators stop oscillating. This makes all the peripheral modules that require an operating clock to stop. Therefore, only the I/O ports can restart the CPU from SLEEP mode (detailed later).

System clock

- Selecting the clock source (OSC module)
  Either OSC3 or OSC1 can be selected as the system clock source. If the application can process the task with a low-speed clock, select OSC1 as the system clock source to reduce current consumption.

- Disabling the OSC3 oscillator circuit (OSC module)
  Enable the oscillator configured as the system clock source and disable another oscillator if possible. Using OSC1 for the system clock and disabling the OSC3 oscillator circuit achieves more reduction of current consumed.

CPU clock (CCLK)

- Executing the `halt` instruction
  Execute the `halt` instruction if there is no task to be processed by the CPU such as when the display on the LCD is only required or when the CPU is waiting an interrupt. Although the CPU enters HALT mode and stops operating, the peripheral modules keep the status when the `halt` instruction is executed. So the LCD driver and the peripheral modules used to generate an interrupt can be made to be run. Power saving effect will be enhanced by disabling the unnecessary oscillator and peripheral modules before executing the `halt` instruction. The CPU reactivates from HALT mode by an interrupt from the ports or peripheral modules that are being operated in HALT mode.

- Selecting a low clock gear (CLG module)
  The CLG module provides clock gears to set the CPU clock speed to 1/1 to 1/8 of the system clock. By running the CPU with the lowest speed required for the application's task, current consumption can be reduced.

Peripheral clock (PCLK)

- Disabling PCLK (CLG module)
  The PCLK supply can be disabled if all the peripheral modules listed below can be placed in standby state.

  Peripheral modules that operate with PCLK
  - Prescaler (PWM & capture timer, remote controller, P port)
  - UART
  - 8-bit timer
  - 16-bit timer Ch.0–2
  - Interrupt controller
  - SPI
  - I^{2}C
  - SVD circuit
  - Power control circuit
  - P port & port MUX (control registers and chattering filters)
  - PWM & capture timer
  - MISC register
  - Remote controller

The peripheral modules listed below operate with a clock other than PCLK except for accessing their control registers. Therefore, PCLK is not required after the control registers are set once and the module starts operating.
- Clock timer
- Stopwatch timer
- Watchdog timer
- 8-bit OSC1 timer
- LCD driver

While PCLK is stopped, no maskable interrupt can be generated. Maskable interrupts are put on hold until PCLK supply is resumed.

Table C.1.1 lists the clock control conditions and how to suspend/resume the CPU operation.

Table C.1.1 List of Clock Control Conditions

Current consumption	OSC1	OSC3	CPU (CCLK)	PCLK peripherals	OSC1 peripherals	CPU suspending method	CPU resuming method
↑ Low	Stop	Stop	Stop	Stop	Stop	`slp` instruction	1
	Oscillating (System clock)	Stop	Stop	Stop	Run	`halt` instruction	1, 2
	Oscillating (System clock)	Stop	Stop	Run	Run	`halt` instruction	1, 2, 3
	Oscillating (System clock)	Stop	Run (1/1)	Run	Run		
	Oscillating	Oscillating (System clock)	Stop	Run	Run	`halt` instruction	1, 2, 3
	Oscillating	Oscillating (System clock)	Run (low gear)	Run	Run		
High ↓	Oscillating	Oscillating (System clock)	Run (1/1)	Run	Run		

Clearing HALT and SLEEP modes (CPU resuming methods)

1. Resuming by a port
   The CPU resumes operating by occurrence of a cause of I/O port interrupt or a debug interrupt (issuing an ICD forced break). If the interrupt controller or the IE flag in the CPU has been set to disable the I/O port interrupt, the CPU does not accept the interrupt request and starts executing the instructions that follow the `halt` or `slp` instruction. When the interrupt has been enabled and PCLK was activated before the `halt` or `slp` instruction is executed, the CPU executes the interrupt handler. When PCLK was stopped before the `halt` or `slp` instruction is executed, the interrupt is put on hold until PCLK supply is resumed even if the interrupt has been enabled.

2. Resuming by an OSC1 peripheral
   The CPU resumes operating by occurrence of a cause of clock timer, stopwatch timer, watchdog timer, or 8-bit OSC1 timer interrupt. If the interrupt controller or the IE flag in the CPU has been set to disable these interrupts, the CPU does not accept the interrupt request and starts executing the instructions that follow the `halt` instruction. When the interrupt has been enabled and PCLK was activated before the `halt` instruction is executed, the CPU executes the interrupt handler. When PCLK was stopped before the `halt` instruction is executed, the interrupt is put on hold until PCLK supply is resumed even if the interrupt has been enabled.

3. Resuming by a PCLK peripheral
   The CPU resumes operating by occurrence of a cause of interrupt in a PCLK peripheral whose interrupt is enabled by the interrupt controller. If the IE flag in the CPU has been set to 0, the CPU does not accept the interrupt request and starts executing the instructions that follow the `halt` instruction. If the IE flag has been set to 1, the CPU executes the interrupt handler.

C.2 Power Saving by Power Supply Control

The following shows some power control methods effective for power saving.

Internal logic voltage regulator

- Setting the internal operating voltage V_{D1} to 2.5 V increases current consumption.
  Set V_{D1} to 1.8 V during normal operation and do not set it to 2.5 V except for Flash programming.
- Enabling the heavy load protection function for the internal logic voltage regulator increases current consumption.
  Disable the heavy load protection function during normal operation except when the internal logic voltage regulator becomes unstable due to driving a heavy load.

LCD system voltage regulator

- Turning the power voltage booster on increases current consumption.
  If the supply voltage V_{DD} is 2.5 V or more, turn the power voltage booster off and drive the LCD system voltage regulator with V_{DD}. The power voltage booster should be used when the supply voltage V_{DD} is less than 2.5 V.
- Enabling the heavy load protection function for the LCD system voltage regulator increases current consumption.
  Disable the heavy load protection function during normal operation except when the display quality on the LCD becomes unstable due to driving a heavy load.
- When the LCD display is not necessary, turn the LCD driver off. Also the power voltage booster should be turned off.

Supply voltage detector (SVD)

- The SVD operation increases current consumption.
  Turn the SVD module off when supply voltage detection is not necessary.

Appendix D Precautions on Mounting

The following shows the precautions when designing the board and mounting the IC.

Oscillator Circuit

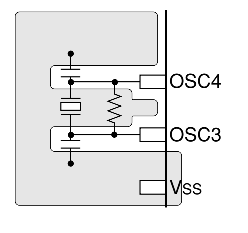
- Oscillation characteristics change depending on conditions such as components used (resonator, R_f, C_G, C_D) and board pattern. In particular, when a ceramic or crystal resonator is used, evaluate the components adequately under real operating conditions by mounting them on the board before the external register (R_f) and capacitor (C_G, C_D) values are finally decided.
- Disturbances of the oscillation clock due to noise may cause a malfunction. To prevent this, the following points should be taken into consideration. In particular, the latest devices are more sensitive to noise, as they are more finely processed.
  The measures against noise for the OSC2 pin, and the components and lines connected to this pin is most essential, and similar measures must also be taken for the OSC1 pin. The measures for the OSC1 and OSC2 pins are described below.
  We recommend taking measures similar to those for the high-speed oscillation system, including the OSC3 and OSC4 pins and the components and lines connected to these pins.

(1) Components that are connected to the OSC1 (OSC3) and OSC2 (OSC4) pins, such as resonators, resistors, and capacitors, should be connected in the shortest line.

(2) Whenever possible, configure digital signal lines with at least three millimeters clearance from the OSC1 (OSC3) and OSC2 (OSC4) pins and the components and lines connected to these pins. In particular, signals that are switched frequently must not be placed near these pins, components, and lines. The same applies to all layers on the multi-layered board as the distance between the layers is around 0.1 to 0.2 mm.
Furthermore, do not configure digital signal lines in parallel with these components and lines when arranging them on the same or another layer of the board. Such an arrangement is strictly prohibited, even with clearance of three millimeters or more. Also, avoid arranging digital signal lines across these components and signal lines.

(3) Shield the OSC1 (OSC3) and OSC2 (OSC4) pins and lines connected to those pins as well as the adjacent layers of the board using V_{SS}.
As shown in the figure on the right, shield the wired layers as much as possible.
Whenever possible, make the whole adjacent layers the ground layers, or ensure there is adequate shielding to a radius of five millimeters around the above pins and lines. As described in (2), do not configure digital signal lines in parallel with components and lines even if such precautionary measures are taken, and avoid configuring signal lines that are switched frequently across components and lines on other layers.

Sample V_{SS} pattern (OSC3)



(4) After taking the above precautions, check the output clock waveform while operating the actual application program in the actual device.
To do this, measure the output of the FOUT1/FOUT3 pins with an oscilloscope.
Check the waveform quality at the OSC3 output clock by measuring the FOUT3 output. Ensure that the frequencies are as designed and that there is no noise or jitters.
Check the waveform quality at the OSC1 clock by measuring the FOUT1 output. Scale up the ranges around the rising and falling edges of the clock pulse to ensure that there is no noise, such as clock and spike, in the 100 ns ranges.

If conditions (1) to (3) are not satisfied, the OSC3 output may be jittery and the OSC1 output may be noisy. When the OSC3 output is jittery, the operating frequency will be lowered. When the OSC1 output is noisy, operation of the timers using the OSC1 clock and the CPU core after the system clock is switched to OSC1 will be unstable.

Reset Circuit

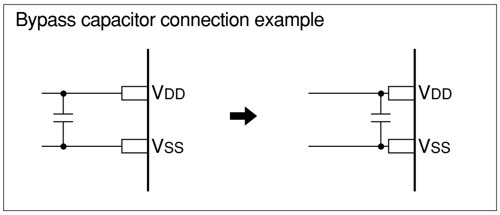
- The power-on reset signal which is input to the #RESET pin changes depending on conditions (power rise time, components used, board pattern, etc.). Decide the time constant of the capacitor and resistor after enough tests have been completed with the application product. The #RESET pull-up resistor should be determined in consideration of the unevenness of the resistor value.
- In order to prevent any occurrences of unnecessary resetting caused by noise during operating, components such as capacitors and resistors should be connected to the #RESET pin in the shortest line.

Power Supply Circuit

- Sudden power supply variation due to noise may cause malfunction. Consider the following points to prevent this:

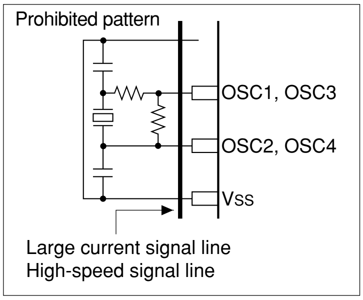
  (1) The power supply should be connected to the VDD and VSS pins with patterns as short and large as possible.

  (2) When connecting between the VDD and VSS pins with a bypass capacitor, the pins should be connected as short as possible.



Arrangement of Signal Lines

- In order to prevent generation of electromagnetic induction noise caused by mutual inductance, do not arrange a large current signal line near the circuits that are sensitive to noise such as the oscillator unit.
- When a signal line is parallel with a high-speed line in long distance or intersects a high-speed line, noise may generated by mutual interference between the signals and it may cause a malfunction. Do not arrange a high-speed signal line especially near circuits that are sensitive to noise such as the oscillator unit.



Noise-Induced Erratic Operations

If erratic IC operations appear to be attributable to noise, consider the following three points.

(1) DSIO pin
Exposure of this pin to low-level noise causes the IC to enter debug mode. In debug mode, the clock is output from the DCLK pin and the DST2 pin is high, indicating that the IC is in debug mode.
In product versions, it is recommended that the DSIO pin be pulled high by connecting it directly to VDD or through a resistor of 10 kΩ or less.
Although the IC contains internal pull-up resistors, it is susceptible to noise because these resistors are high impedance (approximately 100 to 500 kΩ).

(2) #RESET pin
Low-level noise on this pin resets the IC. However, the IC may not always be reset normally, depending on the input waveform.
Due to circuit design, this situation tends to occur when the reset input is in the high state, with high impedance.

(3) VDD and VSS power supplies
If noise lower than the rated voltage enters one of these power-supply lines, the IC may operate erratically.
Take corrective measures in board design; for example, by using solid patterns for power supply lines, adding decoupling capacitors to eliminate noise, or incorporating surge/noise counteracting devices into the power supply lines.

To confirm the above, use an oscilloscope capable of observing higher-frequency waveforms of 200 MHz. The generation of fast noise may not be observed with a low-frequency oscilloscope.

If potential noise-induced erratic operations are detected through waveform observations using an oscilloscope, connect the suspected pin to the GND or power supply with low impedance (1 kΩ or less) and check once again. If erratic operations are no longer detected or occur at reduced frequency, or if different symptoms of erratic operations are observed, said pin may with reasonably certainty be considered to be the source of the erratic operations.

The DSIO and #RESET input circuits described above are designed to detect the edges of the input signal, so that even spike noise may result in erratic operations. Among the digital signal circuits, these pins are most susceptible to noise.
In the design of the circuit board, take the following two points into consideration to protect the signal from noise.

(A) The most important measure is to lower the signal-driving impedance, as described in each item above. Connect pins to the power supply or GND, with impedance of 1 kΩ or less, preferably 0 Ω. In addition, limit the length of the connected signal lines to approximately 5 cm.

(B) Parallel routing of said signal lines with other digital lines on the board is undesirable, since the noise generated when the signal changes from high to low or vice versa may adversely affect signals. The signal may be subject to the most noise when signal lines are laid between multiple signal lines whose states change simultaneously. Take corrective measures by shortening the parallel distance (to several cm) or separating signal lines (2 mm or more).

Precautions for Visible Radiation (when bare chip is mounted)

Visible radiation causes semiconductor devices to change electrical characteristics. It may cause the IC to malfunction or the nonvolatile memory data to be erased. When developing products, consider the following precautions to prevent malfunctions caused by visible radiation.

(1) Design the product and bond the IC on the board so that it is shielded from visible radiation in actual use.

(2) The inspection process of the product needs an environment that shields the IC from visible radiation.

(3) Shield not only the face of the IC but the back and side as well.

(4) After the shielded package has been opened, the IC chip should be bonded on the board within one week. If the IC chip must be stored after the package has been opened, be sure to shield the IC from visible radiation.

(5) If there is a possibility that heat stress exceeding the reflow soldering condition is applied to the IC in the bonding process, perform enough evaluation of data stored in the nonvolatile memory before the product is shipped.

Other

The 0.25 µm fine-pattern process is employed to manufacture this series of products.
Although the product is designed to meet EIAJ and MIL standards regarding basic IC reliability, please pay careful attention to the following points when actually mounting the chip on a board.
Since all the oscillator input/output pins are constructed to use the internal 0.25 µm transistors directly, the pins are susceptible to mechanical damage during the board-mounting process. Moreover, the pins may also be susceptible to electrical damage caused by such disturbances (listed below) whose electrical strength, varying gradually with time, could exceed the absolute maximum rated voltage (2.5 V) of the IC:

(1) Electromagnetic induction noise from the utility power supply in the reflow process during board-mounting, rework process after board-mounting, or individual characteristic evaluation (experimental confirmation), and
(2) Electromagnetic induction noise from the tip of a soldering iron

Especially when using a soldering iron, make sure that the IC GND and soldering iron GND are at the same potential before soldering.

Appendix E Initialize Routine

This section shows a sample vector table and an initialize routine.

boot.s

```
.org    0x8000
.section .rodata                                                           ...(1)

; ======================================================================
;       Vector table
; ======================================================================
                        ; interrupt  vector  interrupt
                        ; number     offset  source

.long BOOT              ; 0x00       0x00    reset                         ...(2)
.long unalign_handler   ; 0x01       0x04    unalign
.long nmi_handler       ; 0x02       0x08    NMI
.long int03_handler     ; 0x03       0x0c    -
.long p0_handler        ; 0x04       0x10    P0 port
.long p1_handler        ; 0x05       0x14    P1 port
.long swt_handler       ; 0x06       0x18    SWT
.long ct_handler        ; 0x07       0x1c    CT
.long t8osc1_handler    ; 0x08       0x20    T8OSC1
.long int09_handler     ; 0x09       0x24    SVD
.long lcd_handler       ; 0x0a       0x28    LCD
.long t16e_handler      ; 0x0b       0x2c    T16E
.long t8f_handler       ; 0x0c       0x30    T8F
.long t16_0_handler     ; 0x0d       0x34    T16 ch0
.long t16_1_handler     ; 0x0e       0x38    T16 ch1
.long t16_2_handler     ; 0x0f       0x3c    T16 ch2
.long uart_handler      ; 0x10       0x40    UART
.long remc_handler      ; 0x11       0x44    REMC
.long spi_handler       ; 0x12       0x48    SPI
.long i2c_handler       ; 0x13       0x4c    I2C
.long int14_handler     ; 0x14       0x50    -
.long int15_handler     ; 0x15       0x54    -
.long int16_handler     ; 0x16       0x58    -
.long int17_handler     ; 0x17       0x5c    -
.long int18_handler     ; 0x18       0x60    -
.long int19_handler     ; 0x19       0x64    -
.long int1a_handler     ; 0x1a       0x68    -
.long int1b_handler     ; 0x1b       0x6c    -
.long int1c_handler     ; 0x1c       0x70    -
.long int1d_handler     ; 0x1d       0x74    -
.long int1e_handler     ; 0x1e       0x78    -
.long int1f_handler     ; 0x1f       0x7c    -

; ======================================================================
;       Program code
; ======================================================================

.text                                                                      ...(3)
.align 1

BOOT:
        ; ===== Initialize ============================================

        ; ----- Stack pointer --------------------
        Xld.a   %sp, 0x0f00                                                ...(4)

        ; ----- Memory controller ----------------
        Xld.a   %r1, 0x5320     ; MISC register base address

        ; FLASHC
        Xld.a   %r0, 0x04       ; 1 cycle access, under 3.3 MHz system clock
        ld.b    [%r1], %r0      ; [0x5320] <= 0x04                         ...(5)

        ; SRAMC
        Xld.a   %r0, 0x00       ; 2 cycle access
        ext     0x01
        ld.b    [%r1], %r0      ; [0x5321] <= 0x01                         ...(6)
        ; ----- ITC (interrupt controller) ------
        Xld.a   %r7, 0x4300     ; ITC register base address

        Xld.a   %r0, 0x1010     ; P0, P1 interrupt level & trigger mode
        ext     0x06
        ld      [%r7], %r0      ; [0x4306] <= 0x1010                      ...(7)

        Xld.a   %r0, 0x1010     ; SWT, CT interrupt level & trigger mode
        ext     0x08
        ld      [%r7], %r0      ; [0x4308] <= 0x1010                      ...(7)

        Xld.a   %r0, 0x1010     ; T8OSC1, SVD interrupt level & trigger mode
        ext     0x0a
        ld      [%r7], %r0      ; [0x430a] <= 0x1010                      ...(7)

        Xld.a   %r0, 0x1010     ; LCD, T16E interrupt level & trigger mode
        ext     0x0c
        ld      [%r7], %r0      ; [0x430c] <= 0x1010                      ...(7)

        ; ===== Main routine =========================================
         ...

; ======================================================================
;       Interrupt handler
; ======================================================================

; ----- Address unalign -------------------------
unalign_handler:
        ...

; ----- NMI -----------------------------------
nmi_handler:
        ...
```

(1) Declare a `.rodata` section to locate the vector table in the `.vector` section.

(2) Define addresses of the interrupt handler routines as vectors.
The `int`*`XX`*`_handler` symbols can be used for software interrupts.

(3) Describe the program code in a `.text` section.

(4) Set the stack pointer.

(5) Set the number of access cycles for the Flash controller.
One-cycle access can be specified only when the system clock frequency is 3.3 MHz or lower.
(See Chapter 3, “Memory Map, Bus Control.”)

(6) Set the number of access cycles for the SRAM controller.
(See Chapter 3, “Memory Map, Bus Control.”)

(7) Set the interrupt trigger mode for the peripheral modules listed below to level trigger.
P0 port, P1 port, stopwatch timer, clock timer, 8-bit OSC1 timer, SVD, LCD driver, PWM & capture timer
(See Chapter 6, “Interrupt Controller (ITC).”)

AMERICA

EPSON ELECTRONICS AMERICA, INC.
HEADQUARTERS
2580 Orchard Parkway
San Jose , CA 95131,USA
Phone: +1-800-228-3964 FAX: +1-408-922-0238

SALES OFFICES
Northeast
301 Edgewater Place, Suite 210
Wakefield, MA 01880, U.S.A.
Phone: +1-800-922-7667 FAX: +1-781-246-5443

EUROPE

EPSON EUROPE ELECTRONICS GmbH
HEADQUARTERS
Riesstrasse 15 Muenchen Bayern, 80992 GERMANY
Phone: +49-89-14005-0 FAX: +49-89-14005-110

ASIA

EPSON (CHINA) CO., LTD.
7F, Jinbao Bldg.,No.89 Jinbao St.,Dongcheng District,
Beijing 100005, China
Phone: +86-10-6410-6655 FAX: +86-10-6410-7320

SHANGHAI BRANCH
7F, Block B, Hi-Tech Bldg., 900, Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5522 FAX: +86-21-5423-5512

EPSON HONG KONG LTD.
20/F., Harbour Centre, 25 Harbour Road
Wanchai, Hong Kong
Phone: +852-2585-4600 FAX: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON (CHINA) CO., LTD.
SHENZHEN BRANCH
12/F, Dawning Mansion, Keji South 12th Road,
Hi- Tech Park, Shenzhen
Phone: +86-755-2699-3828 FAX: +86-755-2699-3838

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
14F, No. 7, Song Ren Road,
Taipei 110
Phone: +886-2-8786-6688 FAX: +886-2-8786-6660

EPSON SINGAPORE PTE., LTD.
1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 FAX: +65-6271-3182

SEIKO EPSON CORPORATION
KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, KOREA
Phone: +82-2-784-6027 FAX: +82-2-767-3677

GUMI OFFICE
2F, Grand B/D, 457-4 Songjeong-dong,
Gumi-City, KOREA
Phone: +82-54-454-6027 FAX: +82-54-454-6093

SEIKO EPSON CORPORATION
SEMICONDUCTOR OPERATIONS DIVISION

IC Sales Dept.
IC International Sales Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 FAX: +81-42-587-5117

S1C17701
TECHNICAL MANUAL

SEIKO EPSON CORPORATION
SEMICONDUCTOR OPERATIONS DIVISION

■ **EPSON Electronic Devices Website**

http://www.epson.jp/device/semicon_e/

Document code: 411089902

First Issue November 2007
Revised October 2008 in JAPANⒹ